



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

PŘEDVOLEBNÍ KALKULAČKA

ELECTION CALCULATOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ MAJERECH

VEDOUcí PRÁCE

SUPERVISOR

Ing. JAROSLAV ROZMAN, Ph.D.

BRNO 2021

Zadání bakalářské práce



23324

Student: **Majerech Tomáš**
Program: Informační technologie
Název: **Předvolební kalkulačka**
Election Calculator

Kategorie: Web

Zadání:

1. Nastudujte procedury hlasování v poslanecké sněmovně a veřejně dostupné informace použitelné pro automatické zpracování.
2. Navrhněte web, do kterého se bude moci uživatel přihlásit, zobrazit si hlasování sněmovny a u jednotlivých hlasování "hlasovat" sám. Webová aplikace mu pak zobrazí, jaké strany jsou nejbližší jeho hlasováním a kteří poslanci jsou mu nejbližší. Navrhněte také způsob výběru vhodných hlasování ze všech hlasování ve sněmovně, např. podle rozdílnosti hlasování nebo umožněte uživatelům označovat hlasování, která jsou podle nich důležitá.
3. Navržený web implementujte a otestujte na skupině uživatelů.

Literatura:

- Web poslanecké sněmovny parlamentu České republiky, <https://www.psp.cz/>, [cit. 15.10.2020]

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 11. listopadu 2020

Abstrakt

Cílem této práce je poskytnout jednoduchý a intuitivní nástroj pro ověření podobnosti politických názorů se skutečnými názory politiků. Není zde prostor na populismus, či nepravdy. Data ze skutečných a již provedených hlasování jasně prozradí, zda politik dostal svým předvolebním slibům a hlasoval dle očekávání. Základní funkcí je umožnění uživateli hlasovat u již existujících hlasování z poslanecké sněmovny a následné zobrazení jeho shody s jednotlivými poslanci a stranami. Kromě toho je cílem také umožnit běžným lidem vyjádřit svůj názor, k tomuto účelu má každý registrovaný uživatel možnost přidat komentář, či hodnocení ve formě až pěti hvězdiček, ke každému poslanci. Aplikace také nabízí stručný a přehledný seznam všech poslanců a schůzí v tomto volebním období.

Abstract

The aim of this thesis is to provide an easy and intuitive tool for checking similarity of political views with real politicians. Here is no possibility for populism or lies. The raw data from the real and already held votes will clearly reveal whether the politician kept his pre-election promises and voted as expected. The basic function is to allow the user to vote in existing votes from Chamber of Deputies and the subsequent display of his match percentage with individual politicians and parties. Another aim is to allow ordinary people to express their opinions, for this purpose each registered user has the opportunity to add a comment or stars rating to each Chamber of Deputies member.

Klíčová slova

webová aplikace, web, volby, politika, django, postgresql, docker, HTML, CSS, JavaScript, MVT

Keywords

web app, web, elections, politics, django, postgresql, docker, HTML, CSS, JavaScript, MVT

Citace

MAJERECH, Tomáš. *Předvolební kalkulačka*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

Předvolební kalkulačka

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rozmana Ph.D. Další informace mi poskytl pan Václav Sklenář z IT oddělení Poslanecké Sněmovny. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Tomáš Majerech

9. května 2021

Poděkování

Děkuji svému vedoucímu, panu Ing. Jaroslavu Rozmanovi, Ph.D. za zajímavé téma a odborné rady v průběhu jeho zpracování. Děkuji své rodině za tichou, ale stabilní podporu po celou dobu studia a děkuji sám sobě, protože si to zasloužím.

Obsah

1	Úvod	3
2	Prerekvizity	4
2.1	Základní pojmy	4
2.2	Existující řešení	6
2.3	Výchozí data	11
3	Použité technologie	13
3.1	HTML, CSS	13
3.2	Bootstrap	13
3.3	Sass	14
3.4	JavaScript	14
3.5	jQuery	14
3.6	Python	15
3.7	Django	15
3.8	Cookiecutter	15
3.9	Docker	16
3.10	Redis	16
3.11	Traefik	16
3.12	PostgreSQL	17
3.13	Další technologie	17
4	Návrh aplikace	18
4.1	Cílová skupina	18
4.2	Případy užití	19
4.3	Návrh databáze	20
4.4	Drátový model	20
4.5	Uživatelské rozhraní	21
4.6	Výběr otázek	22
4.7	Způsob zpracování výsledků kalkulačky	24
4.8	Výběr vhodného serveru	24
4.9	Doména	25
5	Implementace	27
5.1	Struktura aplikace	27
6	Testování	39
6.1	Průběžné testy	39

6.2	Testování produkční aplikace	40
7	Závěr	44
	Literatura	45
A	Instalační manuál	48
A.1	Lokální vývojové prostředí	48
A.2	Produkční prostředí	50

Kapitola 1

Úvod

Každé čtyři roky stojí obyvatelé České republiky před rozhodnutím, komu dají svůj hlas ve volbách do Poslanecké sněmovny. Velké množství z nich nemá mnohdy čas ani chuť pravidelně sledovat dění na politické scéně a nejsou si tudíž jistí, kdo z kandidátů či stran by dostatečně dobře zastupoval jejich názory a zájmy. Proto stále častěji lidé vyhledávají volební kalkulačky, kde můžou pomocí odpovědí na pár otázek zjistit shodu s některým z kandidátů nebo s některou ze stran. Na tomto již existujícím principu staví také tato bakalářská práce a dále ho rozšiřuje a modernizuje.

Skutečnost, že lidé volební kalkulačky pravidelně využívají, skýtá prostor k potenciálnímu budoucímu využití této práce širokou veřejností a její existence by tedy nekončila ve školním archivu, nýbrž by se mohla stát univerzálním nástrojem, usnadňujícím život řadě běžných lidí. To byl také jeden z důvodů, proč mi toto téma dávalo smysl.

Ve druhé kapitole (2) je čtenář seznámen s problematikou legislativního procesu v České republice, strukturou Poslanecké sněmovny a základními pojmy nutnými k porozumění této práci. Také je zde popsána existující konkurence (v době psaní této práce) a zdroj dat. Třetí kapitola (3) rozebírá použité technologie. Kromě HTML a CSS, které se nachází prakticky na každém webu, jsem se rozhodl pro tuto práci využít na klientské části framework¹ Bootstrap a css preprocessor SASS. Na serverové(backend) části je využit python framework Django, k uložení dat je využita objektově-relační databáze PostgreSQL s podporou technologie Redis, zajišťující redukci počtu nutných databázových výpočtů pomocí jejich ukládání do mezipaměti. Na serveru je celá aplikace nasazena v Docker kontejnerech. Ve čtvrté kapitole (4) je popsán postup návrhu webového rozhraní a struktury uložených dat, následovaná kapitolou implementace (5), rozebírající nejdůležitější části vývoje. Výčet kapitol uzavírá kapitola číslo 6, shrnující uživatelské testování a závěr práce (7).

¹Framework: aplikační rámec

Kapitola 2

Prerekvizity

V úvodu této kapitoly jsou stručně shrnuty základní pojmy a procesy týkající se zaměření této práce, zejména z oblasti legislativního procesu. V další části jsou zmíněny již existující konkurenční řešení a jejich nedostatky. Následuje popis primárního zdroje dat.

2.1 Základní pojmy

2.1.1 Poslanecká sněmovna

Poslanecká sněmovna je jedna ze dvou složek, tzv. komor, Parlamentu České republiky. Má 200 členů, tzv. poslanců, volených v přímé volbě na 4 roky [25].

Smyslem Poslanecké sněmovny je tvorba a přijímání zákonů. Návrhy zákonů mohou přijít od poslanců, jejich skupin, krajských zastupitelstev, senátu a vlády [28].

Schvalování zákonů má 3 části zvané čtení.

1. čtení:

Rozhoduje se o tom, jestli se PS bude návrhem vůbec zabývat. Pokud projde, dochází k dalšímu hlasování o tom, jakému výboru bude návrh přiřazen k přezkoumání. V některých případech může být návrh schválen PS už v této fázi, pokud o to navrhovatel požádá.

2. čtení:

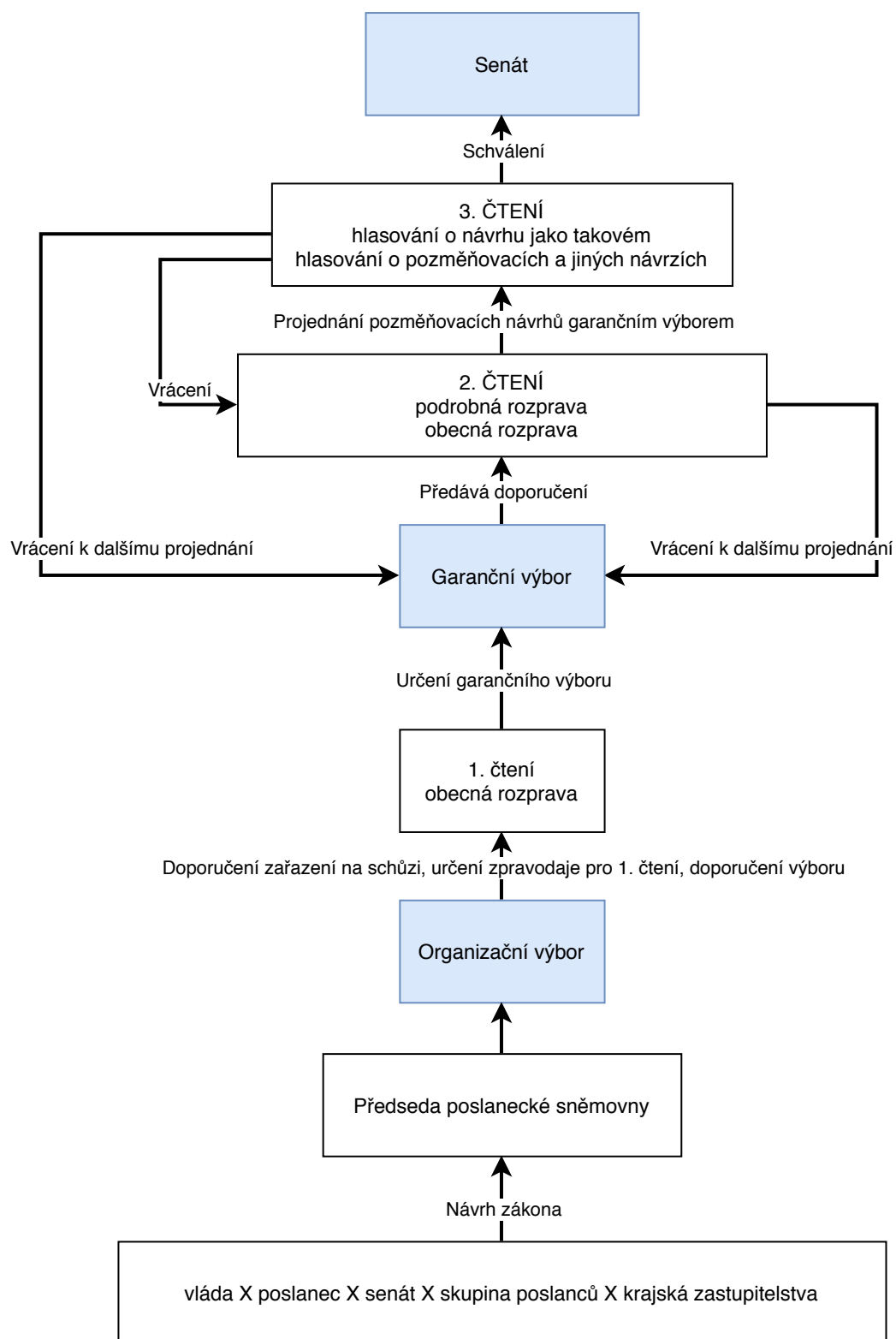
Na začátku vystoupí jeden zpravodaj z každého přiřazeného výboru a informuje PS zprávou a doporučením, na kterém se výbor usnesl. Také se přednáší pozměňovací návrhy.

3. čtení:

Hlasuje se o pozměňovacích návrzích ze 2. čtení, případně o jiných návrzích. Následuje hlasování o přijetí zákona jako celku.

Po přijetí pokračuje návrh do senátu, což je druhá komora Parlamentu České republiky. Ten může návrh odmítnout a vrátit zpět do PS. V takovém případě musí PS o návrhu opět hlasovat a pokud je návrh schválen nadpoloviční většinou všech poslanců (101 hlasů), je senát přehlasován. Posledním krokem je podpis prezidenta České republiky. Ten má ale opět právo návrh vetovat. Zde je postup stejný jako u veta senátního a PS ho může opět přehlasovat nadpoloviční většinou všech poslanců.

Zjednodušený průběh legislativního procesu je znázorněn na obrázku 2.1.



Obrázek 2.1: Zjednodušený graf legislativního procesu v Poslanecké sněmovně

2.1.2 Výbory poslanecké sněmovny

Výbory poslanecké sněmovny jsou skupiny lidí zaměřené na určitý odborný problém, složené jak z členů vlády, tak opozice. Tyto skupiny se návrhem zákona zabývají v odborné rovině a následně na 2. čtení PS doporučí návrh přijmout, zamítnout, nebo navrhnou změny.

Jeden návrh může být přiřazen více výborům, každý z výborů má svého zpravodaje, ale jen jeden z nich je tzv. garanční - ručí za přidělený návrh.

2.1.3 Jednání PS

Jednání, na kterých jsou návrhy zákonů projednávány, se nazývají schůze Poslanecké sněmovny. Aby byla sněmovna usnášeníschopná, musí být přítomno alespoň třetinové kvórum, což odpovídá 67 poslancům [27].

Průběh jednání z hlediska hlasování vypadá následovně:

- Přihlášení účastníků
- Určení zapisovatelů a hlasování o jejich schválení
- Hlasování o jednotlivých bodech aktuální schůze
 - v případě, že vůbec dojde k jejich projednání
- Hlasování o pořadí projednávaných bodů
- Hlasování o konkrétních bodech na pořadu
 - může dojít ke zpochybnění hlasování a k jeho opakování
- V průběhu schůze může dojít také k některým dalším, méně zajímavým, hlasováním například o odložení projednávání nějakého aktuálního bodu na později, či vyhlášení přestávky.

2.1.4 Kvórum

Kvórum je minimální počet hlasů nutný pro přijetí návrhu hlasováním. Běžně je kvórum nadpoloviční většina přítomných poslanců, v některých případech se však může lišit. Například kvórum nadpoloviční většiny všech poslanců nutné pro přehlasování veta Senátu, či prezidenta, nebo kvórum 3/5 přítomných nutné pro přijetí ústavních zákonů [29].

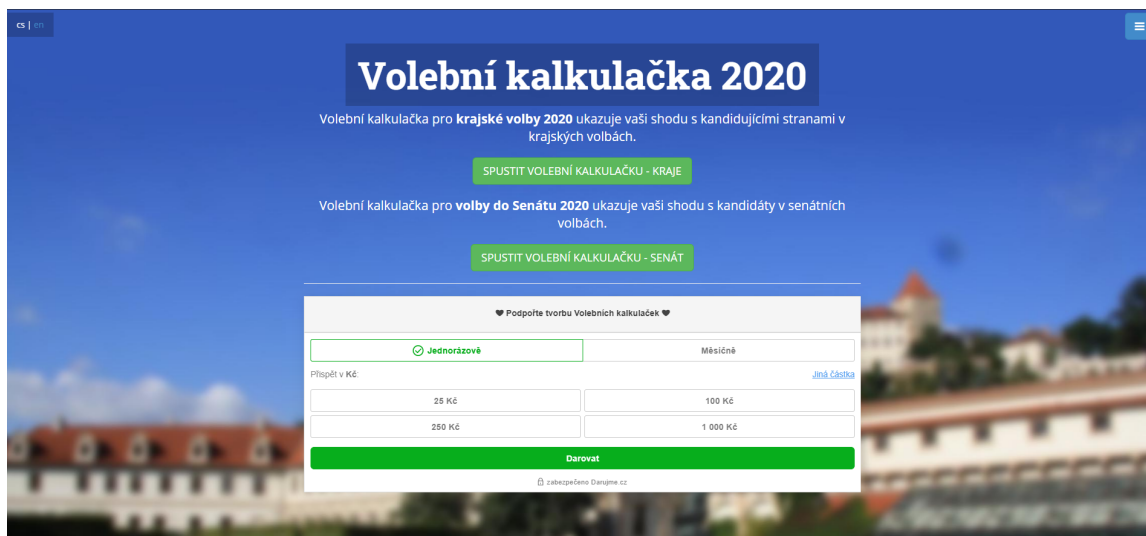
2.1.5 Sněmovní tisky

Sněmovní tisky jsou, zjednodušeně řečeno, dokumenty předkládané poslancům jako podklady pro jednotlivé body programu.

2.2 Existující řešení

Na internetu lze v době psaní této práce nalézt více verzí českých i zahraničních volebních kalkulaček. Jelikož jsou volby obecně pro jednotlivé země silně specifické, zahraniční aplikace zde nebudu rozebírat. Z těch českých jsou nejzajímavější hlavně volebnikalkulacka.cz¹

¹volebnikalkulacka.cz: <https://volebnikalkulacka.cz/>



Obrázek 2.2: Úvodní stránka aplikace volebnikalkulacka.cz

a ta na zpravodajském portálu iDnes.cz². Za zmínku stojí také Postvolítko³, projekt studentů Masarykovy univerzity.

2.2.1 volebnikalkulacka.cz

Princip této volební kalkulačky je jednoduchý. Autor, či autoři, vytvoří soubor otázek, který by podle jejich názoru mohl uživatele zajímat. Tyto otázky jsou následně rozeslány všem zúčastněným politickým stranám, či kandidátům. Na základě matematických modelů jsou poté odpovědi zpracovány a vybrány nejlepší otázky. Tyto jsou nakonec prezentovány v samotné kalkulačce (obr. 2.3), kde na ně odpovídá uživatel. Po ukončení dotazníku je uživateli představena jeho procentuální shoda v odpovědích s jednotlivými stranami či kandidáty (obr. 2.4) [15].

Výhoda tohoto systému je v jeho jednoduchosti a přehlednosti. Otázky bývají jednoznačné, srozumitelné a týkají se aktuálních témat, takže běžný uživatel nemusí mít nic než obecný přehled o hlavním dění v České republice a ve svém kraji.

Za jeho nevýhodu by se dala považovat závislost na pravdivých odpovědích a nepopulistickém přístupu. Může se totiž stát, že kandidát vyplní dotazník podle toho co zrovna v tu chvíli jeho potenciální voliči požadují a následně ve skutečných hlasováních v Poslanecké sněmovně hlasovat jinak.

2.2.2 iDnes.cz

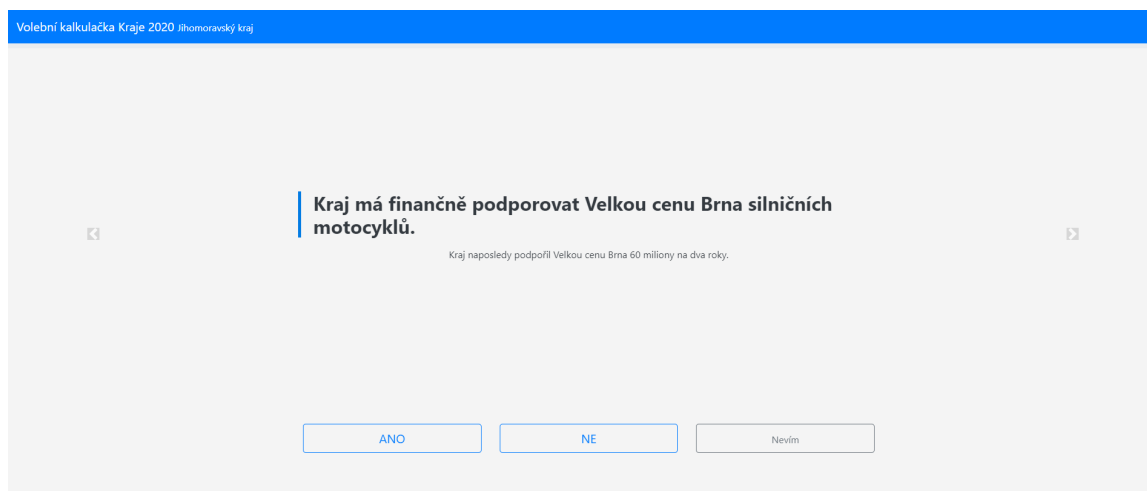
Princip této aplikace je totožný jako u volebnikalkulacka.cz. Liší se pouze některé otázky a grafický vzhled, který je zde přizpůsoben celkovému stylu zpravodajského webu idnes.cz⁴, viz obrázek 2.5. Dokonce mají obě aplikace i stejného autora, kterým je sdružení kohovolit.eu⁵.

²Volební kalkulačka na idnes.cz: <https://volby.idnes.cz/volebni-kalkulacka-kraje.aspx>

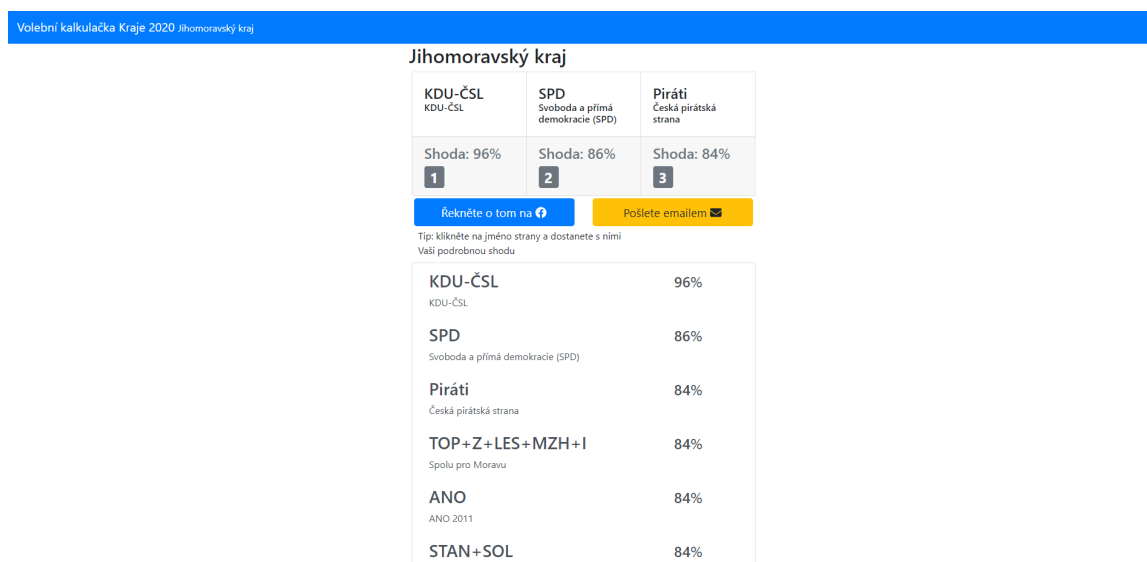
³Postvolítko: <https://www.fi.muni.cz/~xzahrad4/Hlasovatko/index.cgi?o=8>

⁴idnes.cz: <https://www.idnes.cz/>

⁵kohovolit.eu: <http://kohovolit.eu/>



Obrázek 2.3: Příklad otázky v aplikaci volebnikalkulacka.cz



Obrázek 2.4: Příklad prezentace výsledků uživateli v aplikaci volebnikalkulacka.cz

Volební kalkulačka – krajské volby

[KALKULAČKA SENÁT](#)[KALKULAČKA KRAJE](#)[O PROJEKTU](#)

Vyberte si svůj volební obvod z mapy nebo seznamu

--- Zvolte kraj --- ▼



Obrázek 2.5: Úvodní strana volební kalkulačky na webu Idnes.cz

2.2.3 Postvolítka

Přestože nejde o volební kalkulačku, je tato studentská aplikace velmi zajímavý nástroj pro zkoumání dat z Poslanecké sněmovny. Ta jsou zde přehledně a srozumitelně rozřazena do 3 různých přehledů - hlasování, poslanci a strany, jak můžeme vidět na obrázku 2.6.

Přehled hlasování nabízí výčet všech poslaneckých schůzí ve zvoleném volebním období. Tyto schůze jsou dále doplněny o přehled jednacích bodů schůze a k nim náležejících hlasování, obr. 2.7.

U jednotlivých hlasování lze dále otevřít detail s konkrétními jmény poslanců a jejich volbou.

Přehled poslanců nabízí seznam všech aktivních poslanců ve zvoleném volebním období. Pod jménem poslance se dále lze dostat na detail jeho osoby, kde nalezneme kromě jeho veřejně dostupných osobních údajů také statistiky jeho hlasování a absencí.

V přehledu stran nalezneme seznam všech aktuálně zúčastněných stran v poslanecké sněmovně. Přes jednotlivé strany se dá opět dostat na jejich detail. Zde je opět seznam poslanců, pouze omezen na členy konkrétní strany.

Dále zde nalezneme žebříčky podle různých statistik, například podle počtu hlasování pro, počtu neomluvených hodin, či nejnižší absence.

Poslední funkcí Postvolítka je porovnávání dvou vybraných poslanců a zobrazení jejich procentuální shody, vycházející ze stejných voleb u jednotlivých hlasování.



Obrázek 2.6: Úvodní strana webové aplikace Postvolítko



Obrázek 2.7: Přehled schůzí a hlasování PS v aplikaci Postvolítko

Tabulka hl_hlasovani

V souborech uložena jako hlXXXXs.unl, kde XXXX je reference volebního období.

Sloupec	Typ	Použití a vazby
id_hlasovani	int	Identifikátor hlasování
id_organ	int	Identifikátor orgánu, viz organy:id_organ
schuze	int	Číslo schůze
cislo	int	Číslo hlasování
bod	int	Bod pořadu schůze; je-li menší než 1, pak jde o procedurální hlasování nebo o hlasování k bodům, které v době hlasování neměly přiděleno číslo.
datum	date	Datum hlasování
čas	datetime(hour to minute)	Čas hlasování
pro	int	Počet hlasujících pro
proti	int	Počet hlasujících proti
zdrzel	int	Počet hlasujících zdržel se, tj. stiskl tlačítko X
nehlasoval	int	Počet přihlášených, kteří nestiskli žádné tlačítko
prihlaseno	int	Počet přihlášených poslanců
kvorum	int	Kvorum, nejmenší počet hlasů k přijetí návrhu
druh_hlasovani	char(X)	Druh hlasování: N - normální, R - ruční (nejdou známy hlasování jednotlivých poslanců)
vysledek	char(X)	Výsledek: A - přijato, R - zamítnuto, jinak zmatečné hlasování
nazev_dlouhy	char(X)	Dlouhý název bodu hlasování
nazev_kratky	char(X)	Krátký název bodu hlasování

Obrázek 2.8: Popis struktury jedné z datových tabulek Poslanecké sněmovny[19]

2.3 Výchozí data

Data, ze kterých tato práce čerpá, se nachází na oficiálním webu Poslanecké sněmovny Parlamentu České republiky⁶. Konkrétně se jedná o data agend Poslanecké sněmovny⁷, poskytovaná veřejnosti zdarma pod podmínkou uvedení zdroje a data zpracování [19].

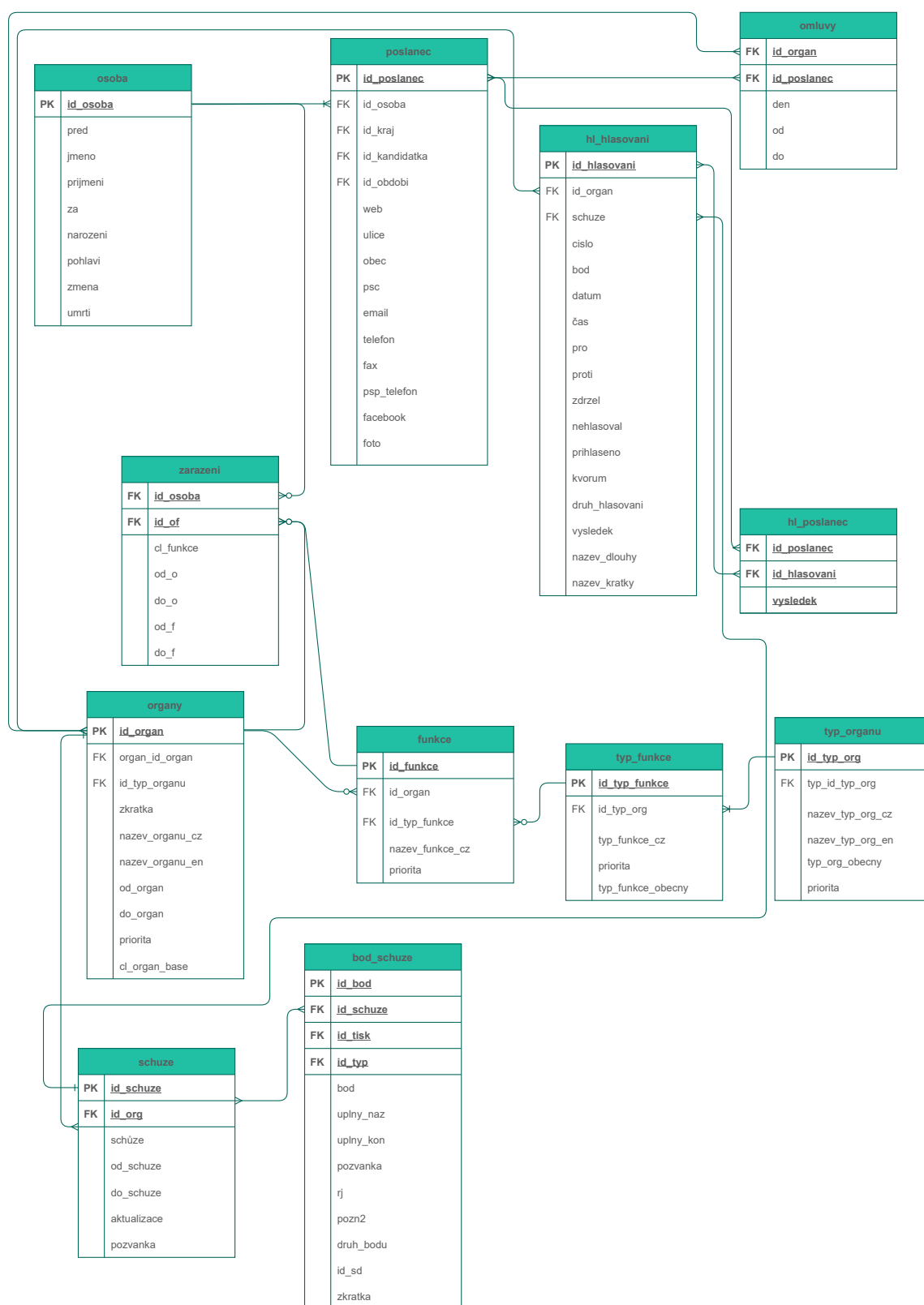
Zde dostupné záznamy popisují všechny schůze Poslanecké sněmovny až do roku 1993. Ty jsou dále děleny na popis samotné schůze a jejího stavu, jednotlivé body pořadu, jejich popis, stav a výsledek. Kromě samotných schůzí jsou k dispozici také záznamy o všech poslancích v tomto volebním období - fotky, pracovní emaily a telefony, ostatní funkce které vykonávají, či jejich hlasování a absence.

Data jsou strukturována v tabulkách, mezi nimiž jsou vzájemné vazby. Na obrázku 2.8 je struktura jedné z databázových tabulek. Ty nejpodstatnější vazby mezi tabulkami pro tuto práci jsou naznačeny na obrázku 2.9.

Jednou denně jsou všechna data aktualizována a doplněna o nové záznamy. Tato aktualizace probíhá v noci.

⁶Oficiální web Poslanecké sněmovny: <https://www.psp.cz/sqw/hp.sqw>

⁷Data agend Poslanecké sněmovny: <https://www.psp.cz/sqw/hp.sqw?k=1300>



Obrázek 2.9: Diagram znázorňující vztahy některých datových tabulek

Kapitola 3

Použité technologie

V této kapitole jsou stručně popsány použité technologie. Jedná se o webovou stránku, proto jsou na úvod představeny základní techniky k vytvoření statického webu. Po nich následuje představení technologií zajišťujících dynamičnost. Zde patří python a jeho framework Django. Pro nasazení na produkčním serveru a mezipaměť (anglicky `cache`) jsou využity technologie Docker¹ a Redis², které propojuje s vnějším světem technologie Traefik³.

Kapitola je zakončena popisem použitého databázového systému PostgreSQL.

3.1 HTML, CSS

HTML je zkratka pro `hyper text markup language`. Jak už název napovídá, tento jazyk není programovací, nýbrž strukturovací (značkovací). To znamená, že pomocí něj dokážeme vytvořit strukturu a význam částí dokumentu, který následně dokáže správně zobrazit a interpretovat webové prohlížeče. Často se přirovnává ke kostře lidského těla [2].

CSS je zkratkou pro `cascading style sheets` a definuje jak budou HTML elementy zobrazeny. V překladu název znamená „kaskádové styly“, kdy kaskádě odpovídá vrstvení jednotlivých stylů na sebe, ovšem vždy platí ten poslední, případně ten nejspecifičtější. Při použití předchozího přirovnání by CSS bylo jako maso a kůže [5].

3.2 Bootstrap

Bootstrap⁴ je aktuálně nejpoužívanější CSS framework pro vývoj responsivních a tzv. `mobile-first` stránek [4]. Poskytuje předpřipravené HTML a CSS komponenty, pomocí kterých se dá rychle a jednoduše vytvořit design webu. Klíčovým aspektem je využívání tříd, kterými je vývojář schopen jednoduše přiřadit HTML elementům styly, které bootstrap již v základu poskytuje, aniž by při tom musel neustále přecházet mezi HTML šablonou a CSS dokumentem. Jednou z nejvyužívanějších funkcí bootstrapu je možnost virtuálního rozdělení obsahu elementu na 12 stejných sloupců, se kterými lze pak jednoduše pracovat pomocí přidávání tříd na elementy obsahu. Velkou výhodou je také jednoduchá práce s responsivitou, opět pomocí různých tříd.

¹Docker: <https://www.docker.com/>

²Redis: <https://redis.io/>

³Traefik: <https://doc.traefik.io/traefik/>

⁴Bootstrap: <https://getbootstrap.com/>

Obecně existují dva způsoby myšlení při vytváření webové stránky. První je navrhnout nejprve verzi pro stolní počítače, respektive celkově velké obrazovky a pak ji postupně obírat o funkcionalitu, aby byla použitelná i na telefonu. Tento způsob byl přijatelný dříve, kdy mobilní telefony sloužily primárně k telefonování a počet návštěvníků webu z nich nebyl nijak závažný. Se zvyšujícím se využitím mobilních telefonů pro prohlížení internetu, kdy v dnešní době více než polovina lidí tzv. „surfuje“ na telefonu [1], je však již považován za nedostatečný.

Tím druhým je právě **mobile first**, kdy se nejprve navrhne design pro mobilní telefony a až s přibývajícím místem na obrazovce se doplňují i pokročilejší designové prvky a funkce. Tento způsob je v současné době obecně využívanější, jelikož je pro vývojáře ve většině případů mnohem komfortnější a zaměřuje se primárně na jednoduchost a ovladatelnost právě na malých obrazovkách, kde to mnohdy může být pro uživatele problém.

3.3 Sass

Sass⁵ je jeden z několika běžně používaných CSS preprocesorů⁶. Ve své podstatě je CSS preprocesor nástroj, respektive jazyk, který zjednodušuje, zpřehledňuje a zrychluje aplikaci CSS webových stránek. Snaží se zvýšit efektivitu psaní kódu pomocí minimalizace nutnosti kopírovat velké bloky již jednou napsaného kódu, což je konkrétně u CSS běžná věc. Přidává například možnost použití proměnných, zanořování selektorů, využívání mixinů.

Prohlížeče nicméně stále ještě pracují pouze s čistým CSS, proto se do něj jazyk preprocesorů musí překládat před tím, než ho server odešle uživateli.

3.4 JavaScript

JavaScript⁷ je scriptovací jazyk umožňující dynamicky měnit obsah stránky bez nutnosti stránku znovu načíst. Kromě toho umí také spoustu dalších věcí, kupříkladu ovládat multimedia a animovat jednotlivé elementy. Naprostá většina moderních prohlížečů má jeho podporu již vestavěnou, mezi výjimky patří například textový prohlížeč Lynx⁸. Podle dostupných statistik [22] je dnes JavaScript využíván na 95% webových stránek.

3.5 jQuery

jQuery⁹ je malá knihovna určená ke zjednodušení práce s JavaScriptem. Hlavní snahou je zmenšení množství napsaného kódu, nutného k vytvoření požadované funkcionality, spolu se zjednodušením používání některých JavaScriptových prvků, kupříkladu AJAX¹⁰ volání [3]. Další z výhod jQuery je obrovské množství dostupných pluginů, díky kterému je často možné jednoduše odstranit i komplikované problémy pomocí již hotového řešení¹¹.

jQuery je v současné době nejspíše nejpopulárnější JavaScriptový framework, využívaný na více než 19 milionech webových stránek [17]. Najdeme ho obsažen také v populárním

⁵Sass: <https://sass-lang.com/>

⁶CSS preprocesory: <https://raygun.com/blog/css-preprocessors-examples/>

⁷JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

⁸Lynx: <https://lynx.browser.org/>

⁹jQuery: <https://jquery.com/>

¹⁰AJAX: Asynchronní JavaScript a XML

¹¹jQuery pluginy: <https://plugins.jquery.com/chosen/>

systému správy obsahu Wordpress¹², nicméně postupně se objevuje snaha od jeho použití upouštět kvůli zbytečnému zvyšování velikosti stránky.

3.6 Python

Python¹³ je interpretovaný, objektově orientovaný, vysokoúrovňový programovací jazyk s dynamickou sémantikou [7].

Kromě toho je v současné době asi nejrychleji rostoucím a nejoblíbenějším programovacím jazykem napříč celou škálou oborů, díky své jednoduchosti a intuitivnosti. Využívá se ve všech možných aplikacích, od základní automatizace opakujících se činností, například zpracování excelu, nebo zpracování dat z webových stránek, přes matematické operace, analýzu a vizualizaci dat, až po vytváření umělé inteligence. Mimo to ho lze využít také k vytváření aplikací - mobilních, desktopových i webových. Právě kvůli možnosti vytvoření webové aplikace, která bude zároveň mít možnost zpracovávat a analyzovat data bez nutnosti přidávání dalších nástrojů, byl jazyk python zvolen pro serverovou část této práce.

3.7 Django

Django¹⁴ je volně dostupný, open-source framework pro webové aplikace napsané v jazyce Python. Jeho smyslem je ulehčit a urychlit vytváření webových aplikací. Za tímto účelem poskytuje již předpřipravenou strukturu, zajišťující například zabezpečení, práci s databází, šablonování, autentizaci a další. Django vychází z návrhového vzoru MVC, což je jeden z nejpopulárnějších návrhových vzorů využívaných pro webové stránky, kde **Controller** je nahrazen šablonami (anglicky „Template“), tedy **MVT**. Kromě toho Django obsahuje množství volně dostupných zásuvných modulů (**pluginů**) rozšiřujících jeho funkcionalitu. Nechybí také přehledná a rozsáhlá dokumentace a široká uživatelská základna nabízející pomoc při řešení problémů.

MVT

MVT, čili **model-view-template** je návrhový vzor rozdělující aplikaci na 3 základní části. Model je databázová část, starající se o to jak jsou data uchovávána, o jejich ukládání a čtení. View obsahuje logiku a komunikuje s Modelem. Template je HTML šablona, které View předá data a kterou django nakonec zobrazí (vyrenderuje) uživateli. Diagram fungování MVT je znázorněn na obrázku 3.1.

3.8 Cookiecutter

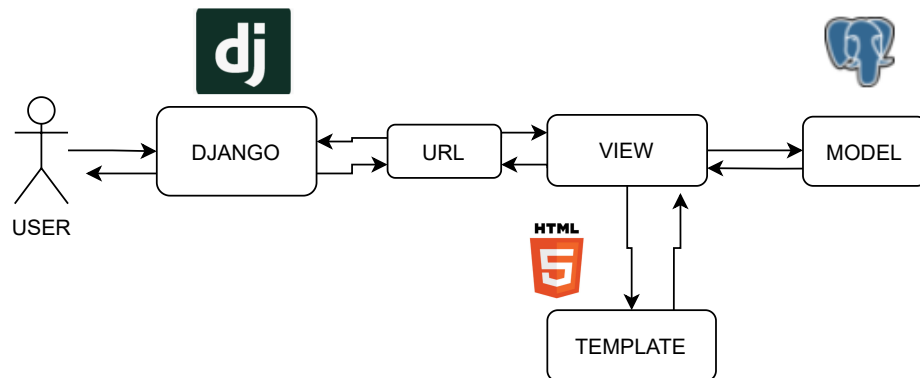
Cookiecutter-Django¹⁵ je generátor základní kostry Django webové aplikace. Uživatel si při spuštění generátoru postupně vybírá jaké technologie chce ve svém projektu použít a aplikace mu podle toho sestaví plně funkční základní moduly a soubory. Díky tomu si projekt udržuje určitý vhodný systém a uspořádání už od počátku.

¹²Wordpress: <https://wordpress.org/>

¹³Python: <https://www.python.org/>

¹⁴Django: <https://www.djangoproject.com/>

¹⁵Cookiecutter: <https://github.com/pydanny/cookiecutter-django>



Obrázek 3.1: Fungování návrhového vzoru MVT

3.9 Docker

Docker je, zjednodušeně řečeno, virtualizační nástroj umožňující umístit aplikaci spolu se všemi jejími potřebami na server nezávisle na jeho druhu a nainstalovaných aplikacích. Docker kontejner poskytuje základní, předpřipravené prostředí, ve kterém aplikace následně běží, podobně jako běžné virtualizované systémy. Jeho výhodou však je, že na rozdíl od nich není tak náročný, je menší a jednodušeji spravovatelný.

Každá služba má svůj vlastní Docker kontejner - jeden kontejner zvlášť pro každou ze služeb Redis, PostgreSQL, Django a Traefik. Tyto mezi sebou mohou komunikovat díky propojení vlastní lokální sítě.

Aby nebylo potřeba ručně při každém zapnutí aplikace konfigurovat každý z kontejnerů zvlášť, existuje `docker-compose`. Správce aplikace sepíše všechno nastavení do souboru zvaného `Dockerfile` a `docker-compose` se po spuštění postará o zbytek.

3.10 Redis

Redis je NoSQL¹⁶ databáze. Běžné SQL databáze využívají k ukládání dat tabulky, které jsou dále členěny na sloupce a řádky, kdy sloupce mají pevně daný typ dat které mohou obsahovat. Redis využívá dvojici klíč:hodnota, kde hodnota může nabývat jeden z několika typů. Je schopen celý pracovat uvnitř paměti, což z něj činí velmi rychlý a efektivní nástroj vhodný pro použití jako cache, kdy díky němu můžeme místo zatěžování serveru opakovanými náročnými operacemi pouze vyčíst již jednou získané výsledky z mezipaměti s velmi nízkými náklady. V současné době je tímto způsobem využíván například na Twitteru, StackOverflow či Pinterestu.

3.11 Traefik

Traefik je open source proxy. Jeho úlohou je v tomto případě mapování vnějších portů na aplikaci. Tzn. pokud někdo přistoupí na adresu serveru na port 80, Traefik tento přístup přepošle (přeloží) na port v django docker kontejneru. Konkrétní port závisí na nastavení v `Dockerfile`.

¹⁶NoSQL: <https://aws.amazon.com/nosql/>

3.12 PostgreSQL

PostgreSQL¹⁷ je výkonný, objektově-relační, open-source, databázový systém využívající a zároveň rozšiřující jazyk SQL. Jeho vznik se datuje až do roku 1986 a za dobu své existence si získal pověst spolehlivého, robustního a jednoduše rozšiřitelného řešení databáze [11].

Při volbě databázového systému byl původní kandidát MySQL, jelikož s ním často pracuji a všechno potřebné jsem tak měl již připravené. Nicméně podle některých názorů [16][8] je obecně lepší využívat databázový systém PostgreSQL, jelikož je na něj Django lépe připravené a používají ho i sami vývojáři, takže bude lépe dokumentován.

3.13 Další technologie

Za zmínku stojí také **npm**¹⁸, což je nejpoužívanější správce balíčků pro jazyk JavaScript [6]. V této práci je využíván zejména na kompilaci jazyka SASS do CSS a automatickou obnovu lokálního serveru.

¹⁷PostgreSQL: <https://www.postgresql.org/>

¹⁸NPM: <https://www.npmjs.com/>

Kapitola 4

Návrh aplikace

U webových aplikací obecně platí, že rozhodnutí uživatele o využití či opuštění stránky z velké části závisí na designu, použitelnosti ale hlavně prvním dojmu. Proto je třeba, aby kromě kvalitní funkcionality měla aplikace také moderní a promyšlený vzhled.

Nejprve je třeba vytyčit cílovou skupinu. Díky tomu zjistíme, na jaké části systému je třeba se zaměřit a případně jakým způsobem je nejlépe zpracovat pro co největší uživatelský komfort. Dalším krokem je analýza případů užití, pomocí které zjistíme jaké funkce budou uživatelé potřebovat a používat. Následuje návrh databáze, vycházející z dat, se kterými chceme pracovat a z jejich vztahů. Dalším bodem je návrh grafické prezentace. Po ní už následují konkrétní teoretická řešení problémů implementace - výběr vhodných otázek, zpracování výsledků a postup při výběru severu a domény.

Kapitola vrcholí samotným grafickým návrhem, zpracovávajícím všechny předchozí kroky do jednoho úhledného a intuitivního uživatelského rozhraní.

4.1 Cílová skupina

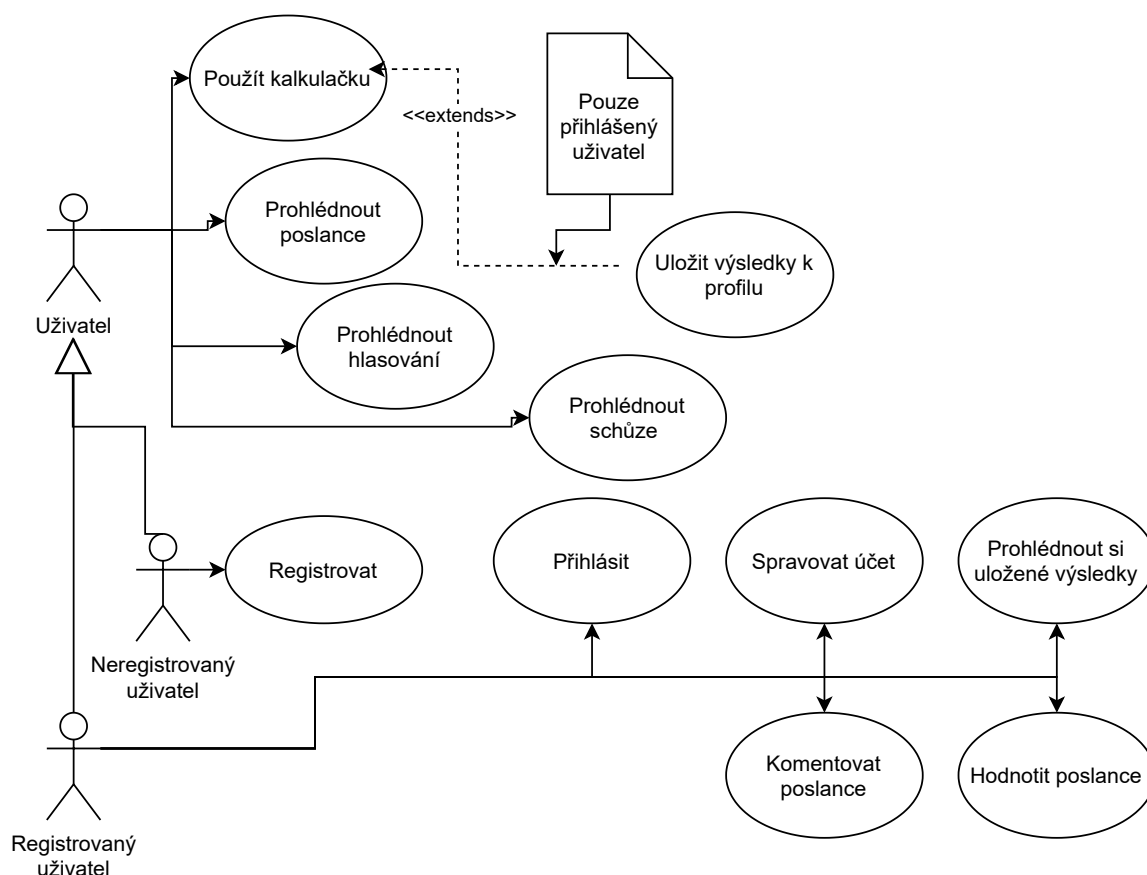
Cílová skupina kalkulačky budou obecně všichni lidé s volebním právem, tedy občané České republiky starší 18 let [26]. Podle statistiky zveřejněné na webu statistikaamy.cz¹, který spravuje Český statistický úřad², používalo v roce 2016 internet 32,5% z lidí starších 65 let [24]. Bude tedy třeba zvýšit důraz na jednoduchost, přehlednost a intuitivnost, které jsou zejména pro starší lidi kritické.

Dále lze tuto skupinu rozdělit na dvě menší části. Tou první jsou lidé, kteří využijí naplno všech funkcí webu, včetně registrace a uložení výsledků na svůj profil, hodnocení poslanců a komentářů. Zde budou spadat zejména mladší uživatelé. Pro ty bude důležité, aby registrace byla jednoduchá a rychlá a zároveň jim web umožňoval spravovat svůj profil a své uložené výsledky.

Zbýlá část bude tvořena lidmi, které bude zajímat pouze jejich výsledek a hned po jeho zjištění web opustí. V tomto případě je nutné co nejvíce zjednodušit přístup k hlavní funkční části webu.

¹statistikaamy.cz: www.statistikaamy.cz

²Český statistický úřad: <https://www.czso.cz/csu/czso/domov>



Obrázek 4.1: Diagram případů užití pro volební kalkulačku

4.2 Případy užití

Po příchodu na web uvidí každý uživatel domovskou stránku, poskytující základní přehled o smyslu a cíli aplikace. Na výběr zde má celkem tři akční oblasti. Kromě horního menu a patičky, které jsou přístupny na každé stránce aplikace, také velké akční tlačítko sloužící pro přechod do sekce s popisem kalkulačky. Každý uživatel bez rozdílu může hlasovat u jednotlivých hlasování a používat kalkulačku. Ovšem pouze registrovaným uživatelům jsou výsledky průběžně ukládány a neztratí se například po vymazání mezipaměti prohlížeče.

Horní menu odkazuje na jednotlivé části aplikace, na přehledy poslanců, hlasování a stran, které může použít každý uživatel. Je zde také tlačítko pro přihlášení, případně odkaz na uživatelský účet, pokud již uživatel přihlášený je. Pokud není, je mu přes něj umožněno se buď přihlásit, nebo přejít dalším odkazem na registrační formulář. Z přehledu poslanců se dá následně přejít na detail jednotlivých poslanců. Zde může registrovaný uživatel ohodnotit konkrétního poslance pomocí hvězdiček, či přidat komentář. Tyto dvě funkce byly omezeny pouze pro přihlášené uživatele z důvodu možného falšování hodnocení, či urážlivých komentářů.

Přihlášeným uživatelům se přechodem do jejich účtu umožní upravovat svůj profil - přidat či upravit profilovou fotku, změnit heslo, změnit svůj email.

V patičce pak má každý uživatel přístup k informacím o webu a ke stránce s kontakty.

Případy užití znázorňuje diagram případů užití na obrázku 4.1

4.3 Návrh databáze

Největší část databáze tvoří tabulky převzaté z výchozích dat. Zde prakticky nedošlo k žádným úpravám vůči jejich popisu na webu Poslanecké sněmovny³. Pro uložení ohodnocení jednotlivých podstatných hlasování bylo nutno přidat tabulku `hl_hlasovani_rating`, uchováající ohodnocení ke každému hlasování. Django potřebuje také tabulky k uchovávání informací o registrovaných uživateliích a jejich oprávněních, či databázových migracích, což jsou zjednodušeně řečeno záznamy o změnách databázové struktury. Dále byly přidány tabulky pro ukládání komentářů a hvězdičkových hodnocení jednotlivých poslanců. Všechny tabulky a jejich vzájemné vztahy jsou znázorněny v diagramu s názvem `model-diagram.svg` ve složce **Diagramy** na paměťovém médiu.

Tabulek je příliš velké množství na popis, nicméně za zmínku stojí hlavně tabulky

- `hl_hlasovani`
uchováající informace o jednotlivých hlasováních
- `hl_poslanec`
uchováající všechna data o hlasování jednotlivých poslanců pro každé z existujících hlasování v předchozí tabulce
- `hist`
která je využita pro získání informací pro zjištění důležitých hlasování
- `organy`
uchováající informace o orgánech, což můžou být strany, skupiny poslanců, volební období, výbory, a podobně
- `zarazeni`
kde nalezneme informace o zařazení jednotlivých poslanců v orgánech
- `osoby`
která uchovává informace o jednotlivých osobách a na kterou se odkazuje tabulka `hl_poslanec`

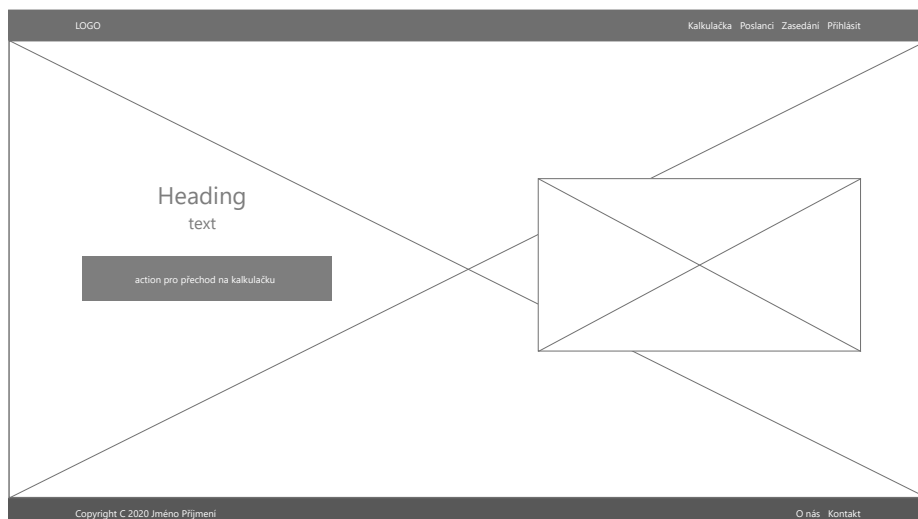
4.4 Drátový model

Drátový model, neboli takzvaně **wireframe**, slouží ke zjednodušené vizualizaci umístění jednotlivých prvků webu v prostoru. Měl by obsahovat všechny elementy, které se budou na výsledném webu nacházet, nejlépe již i v konkrétních velikostech, aby šlo už v této fázi zjistit a odstranit případné nedostatky v rozložení. Smyslem tohoto modelu není zobrazit jak bude web vypadat, pouze jeho strukturu. Proto se používají spíše černobílé, či jednobarevné modely, ve kterých jsou designové bloky a obrázky nahrazeny jednoduchými obdélníky.

V ideálním případě se pro každý typ podstránky dělají až 4 drátové modely, v závislosti na různých typech responzivního zobrazení. V této práci postačuje jedna velikost, vytvořená v programu AdobeXD⁴. Vytvořený drátěný model úvodní strany lze vidět na obrázku 4.2. Kompletní drátové modely jsou přiloženy na datovém médiu ve složce **Grafické návrhy**.

³Data PSP: <https://www.psp.cz/sqw/hp.sqw?k=1301>

⁴AdobeXD: <https://www.adobe.com/products/xd.html>



Obrázek 4.2: Drátěný model úvodní strany aplikace

4.5 Uživatelské rozhraní

Při návrhu uživatelského rozhraní je obecně třeba dodržovat základy UX⁵ (uživatelská zkušenost) a UI⁶ (uživatelské rozhraní) designu. Tyto dva pojmy se v stále častěji zmiňují při tvorbě webových stránek a aplikací, nicméně nezřídka dochází k jejich zaměňování.

UX design řeší, jakým způsobem bude uživatel s produktem pracovat. Tím obecně nemusí být myšlena jen digitální aplikace, ale může se jednat také o nějakou fyzickou věc, či službu. O cokoliiv co může uživatel nějakým způsobem zažít. Do UX designu patří například drátový model, viz sekce 4.4.

Pokud jde o UI, tak na rozdíl od UX se jedná již o čistě digitální pojem. Uživatelské rozhraní je, zjednodušeně řečeno, místo, na kterém uživatel může komunikovat s aplikací. UI zahrnuje celkový vzhled, interaktivitu a intuitivnost aplikace a také celkový uživatelský pocit z používání. V současné době se zde řadí také responsivita.

4.5.1 Postup při návrhu uživatelského rozhraní

Jako první bylo třeba vypracovat základní strukturu webu se všemi prvky. K tomu posloužil již zmiňovaný drátový model. Tento byl po vytvoření prezentován nezainteresovaným lidem v roli uživatelů, aby byla získána zpětná vazba. Na jejím základě došlo k několika úpravám.

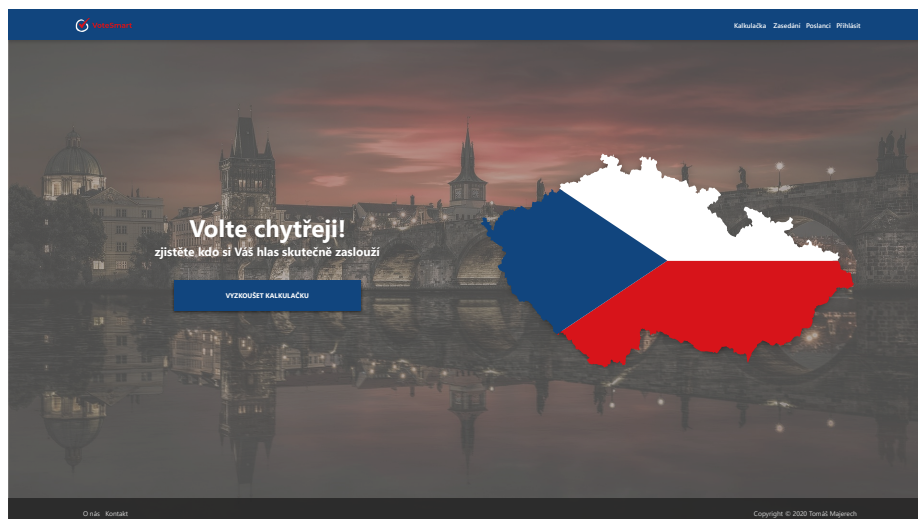
Jako další krok bylo potřeba vymyslet název a z něj vycházející logo celé aplikace. Jelikož smyslem práce je poskytnout lidem radu, čili učinit je chytřejšími (chytrý je anglicky **smart**), a následně jim díky tomu ulehčit výběr při volbě (anglicky **vote**), nabízelo se spojení právě těchto dvou slov. Původní volba byla SmartVote, nicméně po kontrole dostupnosti domény byla zvolena prohozená verze, čili VoteSmart - česky **Volte chytře**. Samotné logo bylo následně vytvořeno ve webové aplikaci [freelogodesign.org](https://www.freelogodesign.org/)⁷. Je tvořeno jménem webu a symbolem zaškrtnutí políčka, symbolizujícím hlavní účel webu - volbu.

Následuje výběr vhodného barevného schématu. Jelikož se jedná o politicky orientovanou aplikaci pro prostředí České republiky, nabízí se použití národních barev - červené,

⁵UX: user experience

⁶UI: user interface

⁷freelogodesign.org: <https://www.freelogodesign.org/>



Obrázek 4.3: Grafický návrh úvodní strany aplikace

modré a bílé. Konkrétní barevné hladiny (#11457e pro modrou a #d7141a pro červenou) byly převzaty z jedné z použitých grafik, konkrétně z mapy České republiky [9], viditelné na grafickém návrhu na úvodní straně.

Jako font byl zvolen Segoe UI. Primární používané velikosti jsou 35px pro nadpisy H1, 25px pro H2 a 16px pro běžný text.

Samotný grafický návrh, umístěn na paměťovém médiu ve složce **Grafické návrhy**, byl vypracováván opět v programu AdobeXD. Na obrázku 4.3 můžete vidět design úvodní strany. Při vytváření grafického návrhu byly použity principy responsivního designu, nicméně samotný návrh je pouze pro desktopovou⁸ verzi.

Při vytváření grafické předlohy byly zjištěny určité nedostatky, na které jsem při vytváření drátového modelu nenarazil. Je tedy v některých částech odlišná. Také došlo k zásadním změnám v principu fungování aplikace oproti původnímu plánu, které v drátovém modelu nejsou zpětně reflektovány.

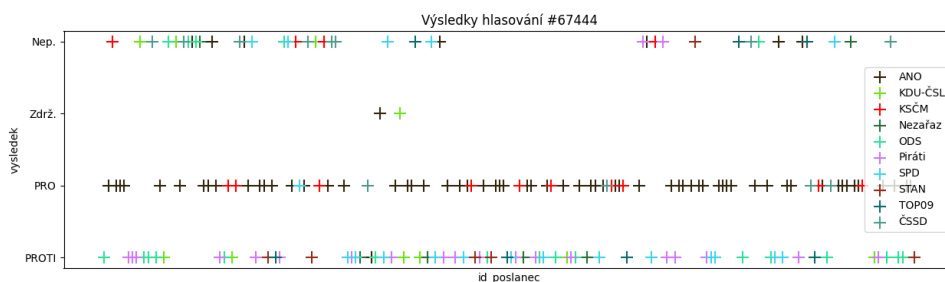
4.6 Výběr otázek

Jedná se o nejobtížnější a nejspíše také nejdůležitější část celé práce. Na výběru vhodných otázek totiž záleží nejenom to, nakolik dokážeme uživateli přiřadit podobnost s jednotlivými stranami a poslanci, ale také zda uživatel vůbec bude chtít aplikaci používat.

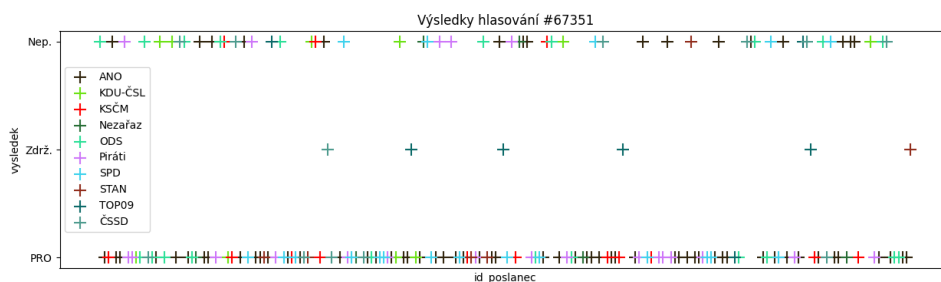
Jelikož uživatel si bude vybírat hlasování, u kterých chce sám hlasovat, je zapotřebí mu jich předložit co největší počet. Nicméně jen za aktuální volební období je těchto hlasování takřka 8000 a naprostá většina z nich nemá ve výsledku žádný vliv na to, zda bude konkrétní návrh zákona přijat či zamítnut. Proto je zde nutné zavést určitou algoritmizaci, usnadňující uživateli práci s aplikací.

Důkladným průzkumem dat, která jsou podkladem této práce, jsem došel k závěru, že na jejich základě není možné u jednotlivých hlasování strojově určit, zda konkrétní hlasování bylo rozhodující například pro přijetí návrhu a jeho posunutí do dalšího kroku, či zda

⁸Desktopová verze je slangový výraz pro běžné obrazovky stolních počítačů, obecně s poměrem stran 16:9



Obrázek 4.4: Vizualizace „přínosného“ hlasování pro kalkulačku



Obrázek 4.5: Vizualizace „nepodstatného“ hlasování pro kalkulačku

šlo kupříkladu pouze o hlasování o výboru, který dostane již přijatý návrh ke zpracování. Ohledem tohoto zjištění jsem kontaktoval přímo správce dat Poslanecké sněmovny, který tento fakt potvrdil. Také však uvedl, že pro některá hlasování je možné tyto informace dohledat prostřednictvím sněmovních tisků. Ty jsou součástí každého návrhu zákona a každá změna jejich stavu je zaznamenávána v tabulce `hist`.

Vzhledem k nemožnosti vyřadit irelevantní hlasování byla tedy využita pouze ta, u kterých lze pomocí tabulky `hist` určit jejich důležitost. Tím se snížil počet prezentovaný uživateli na přibližně 900 za poslední dvě volební období.

Jelikož se stále jedná o příliš velký počet, je třeba uživateli dále usnadnit práci. Konkrétně lze například seřadit hlasování podle relevance dle algoritmického vyhodnocení jejich skóre na základě rozdílnosti hlasování jednotlivých stran. Čím větší různorodost ve hlasování napříč stranami, tím lépe z hlediska určení podobnosti uživatele se stranami a poslanci po vyhodnocení v kalkulačce.

Jakým způsobem je rozdílnost hlasování podstatná pro výpočet korelace s jednotlivými stranami je patrné například z vizualizace na obrázcích 4.4 a 4.5. Zatímco z prvního grafu bychom nedokázali určit prakticky nic, jelikož takřka všichni přítomní hlasovali stejně, z toho druhého dokážeme již alespoň určitou rozdílnost vyčíst. Jen pro pořádek je však třeba zmínit, že podobnost v jednom hlasování není dostatečná a pro dostatečně validní výstupy bude třeba, aby uživatel hlasoval u několika. Čím více hlasování, tím přesnější výsledky.

4.7 Způsob zpracování výsledků kalkulačky

Uživatel bude „hlasovat“ ve stejných hlasováních jako skuteční poslanci. Porovnávání proto bude probíhat komparací jeho voleb s hlasováním jednotlivých poslanců a stran u těch hlasování, kde uživatel uvedl svoji volbu. Tzn. pokud bude uživatel hlasovat například u #67351 a #67444, budou jeho výsledky porovnávány s výsledky poslanců pouze v těchto dvou konkrétních hlasováních. Následně se pro každého poslance vypočte procentuální shoda a výsledky se seřadí od té největší.

Porovnání se stranami záleží na poměru stejných odpovědí členů strany s celkovým počtem členů pro každou uživatelem odhlasovanou otázku zvlášť.

4.8 Výběr vhodného serveru

Aby aplikace mohla být lidmi využívána, musí být umístěna na veřejně přístupném serveru. Zde máme pro web obecně 3 možnosti:

- Sdílený hosting⁹

Jedná se o nejlevnější a obecně nejjednodušší možnou variantu. Daň za nízkou cenu je však ve sdílení hardwaru s dalšími zákazníky. Z toho důvodu není možné dopředu předvídat dostupné prostředky a náročnější aplikace můžou skončit chybou kvůli jejich spotřebování ostatními. Také zde uživatel ve většině případů nemá kontrolu nad nainstalovanými aplikacemi a může využívat pouze omezený počet těch předpřipravených. Je zde také bezpečnostní riziko, protože data různých aplikací jsou na stejných úložištích. Jako příklad můžeme uvést například endora.cz¹⁰

- Virtuální privátní server (VPS)

U tohoto typu hostování je opět hardware serveru sdílen mezi několika aplikacemi. Každý uživatel však již má k dispozici vlastní virtualizovaný systém, který je oddělen od ostatních. Poskytuje tak uživateli mnohem více prostoru pro správu i lepší zabezpečení. Tuto sekci můžeme dále rozdělit na spravované a nespravované virtuální privátní servery. Ty spravované opět uživateli určitým způsobem omezí kontrolu nad nainstalovanými službami, nabídnou však většinou jednoduché možnosti nasazení předem připravených prostředí pro určité specifické zaměření přes přívětivé webové rozhraní, například na cloudways.com¹¹ lze jedním klikem spustit virtualizovaný server pro WordPressovou aplikaci, kdy po pár minutách uživatel dostane již plně funkční prostředí s funkční WordPress aplikací včetně databáze, cache a základních pluginů.

U nespravovaných serverů má uživatel k dispozici pouze samotný server, přístup k němu a pár základních funkcí přes webové rozhraní, například možnost ho vypnout či resetovat. Všechnu další správu si uživatel musí zajistit sám - instalaci potřebných aplikací, nainstalování webového serveru, databáze, jejich propojení, nasměrování domény či vytváření subdomén, emaily a další.

Podstatně dražší, než sdílená verze. Např. zmiňované cloudways.com

- Dedikovaný sever

Nejdražší varianta. Jde o kompletně vyčleněný hardware pro použití jedním zákazní-

⁹Hosting je zažitý název pro server na kterém je aplikace, či webová stránka umístěna

¹⁰endora.cz: <https://www.endora.cz/>

¹¹cloudways.com: <https://www.cloudways.com/en/>

kem. Zde má zákazník plnou kontrolu nad obsahem serveru a nainstalovanými aplikacemi. Například sh.cz¹²

Jelikož framework Django nespadá do úplně běžných případů užití, většina sdílených řešení ho nepodporuje. Jako východisko se tedy nabízí buď využít školní server Eva, nebo sáhnout po nějakém dražším sdíleném.

Zejména z důvodu výhod naprosté kontroly nad systémem a jednoduchosti nasazení jsem zvolil digitalocean.com¹³, kde se dostačující sdílený VPS server dá pořídit za zhruba 5\$ měsíčně.

4.9 Doména

Jelikož přístup na web přes IP adresu není moc praktický, potřebuje server zapamatovatelnou doménu, pod kterou ho lidé budou moci vyhledat. Domény s koncovkou .cz stojí zhruba 200 korun ročně a koupit ji není nic složitého. Například formulář na webu endora.cz¹⁴ je velmi intuitivní. Zde byla také zakoupena doména použitá na vývoj této práce. Po pozdějším průzkumu bych nicméně doporučil pro umístění domén spíše wedos.cz¹⁵, vzhledem k lepším možnostem pro přesměrování domény na data umístěná pod jiným správcem i k celkové roční ceně za tzv. „parkování“. Domény lze však většinou kdykoliv bez poplatku přesunout mezi správci.

Aby však web byl pod touto doménou dostupný, je třeba jí nejprve přidat záznam NS s odkazem na DNS servery¹⁶ zvoleného poskytovatele hostingu. Pro domény s koncovkou .cz je tato akce navíc specifická, protože nelze přidat jednotlivé DNS servery, nýbrž je potřeba vytvořit tzv. NSSET¹⁷ a ten teprve přidat k záznamům pro doménu. Tím bude zajištěno, že každý dotaz hledající tuto URL adresu bude nasměrován správným směrem.

Aby DNS servery uměly hledanou URL adresu přeložit na IP adresu, je třeba na serveru nastavit záznamy typu A(IPV4), AAAA(IPV6) a NS. Toto nastavení lze například pro digitalocean.com vidět na obrázku 4.6.

¹²sh.cz: <https://www.sh.cz/dedikovane-servery>

¹³digitalocean.com: <https://www.digitalocean.com/>

¹⁴Endora.cz: <https://www.endora.cz/>

¹⁵wedos.cz: <https://www.wedos.cz/>

¹⁶Dns server je v podstatě překladač url adresy na IP adresu, viz <https://www.cloudflare.com/learning/dns/what-is-a-dns-server/>

¹⁷Vytvoření NSSETu: <https://objednavka.forpsi.com/domain/cr-nsset.php>

Create new record

A AAAA CNAME MX TXT NS SRV CAA

Use @ to create the record at the root of the domain or enter a hostname to create it elsewhere. A records are for IPv4 addresses only and tell a request where your domain should direct to.

HOSTNAME	WILL DIRECT TO	TTL (SECONDS)	
<input type="text" value="Enter @ or hostname"/>	<input type="text" value="Select resource or enter custom IP"/>	<input type="text" value="Enter TTL 3600"/>	<input type="button" value="Create Record"/>

DNS records

Type	Hostname	Value	TTL (seconds)	
A	www.tomasmajerech.cz	directs to 167.99.254.121	3600	More ▾
A	tomasmajerech.cz	directs to 167.99.254.121	3600	More ▾
NS	tomasmajerech.cz	directs to ns3.digitalocean.com.	1800	More ▾
NS	tomasmajerech.cz	directs to ns1.digitalocean.com.	1800	More ▾
NS	tomasmajerech.cz	directs to ns2.digitalocean.com.	1800	More ▾

Obrázek 4.6: Rozhraní administrace DNS na webu digitalocean.com

Kapitola 5

Implementace

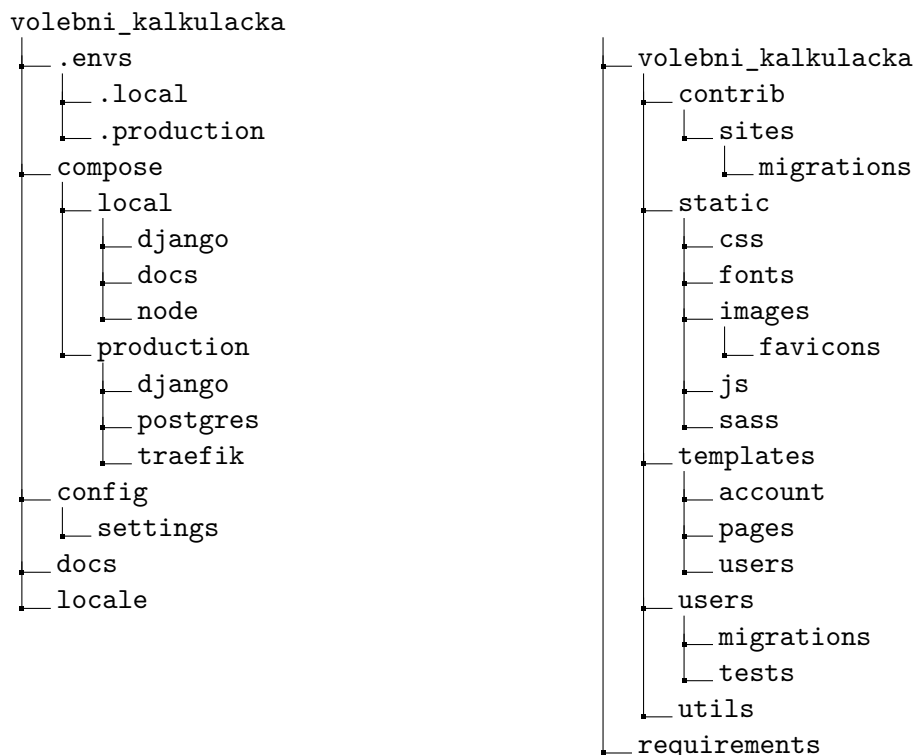
Cílem této kapitoly je poskytnout náhled do důležitých kroků implementace. Nejprve je popsána základní struktura aplikace a import dat včetně nalezených problémů. Následuje popis způsobu ohodnocení významných hlasování a určení shod s poslanci a stranami.

5.1 Struktura aplikace

Jelikož základním programovacím jazykem aplikace je python, je třeba instalovat jeho moduly. Ty lze samozřejmě instalovat i globálně do celého systému, nicméně běžně se pro jejich sdružování k jednotlivým projektům používá tzv. **virtuální prostředí**¹. Tím docílíme oddělení python modulů využitých pro tento projekt od případných jiných verzí jinde v systému. Díky tomu odpadají problémy s možnou nekompatibilitou napříč projekty.

Samotná struktura projektu je vygenerována pomocí nástroje `Cookiecutter-django`. Adresářová struktura vypadá následovně.

¹Python Virtuální prostředí: <https://docs.python.org/3/library/venv.html>



Tato kostra byla poté podle potřeby rozšiřována. Byla přidána samostatná pod-aplikace pro správu importovaných dat `dataImport` a vlastní moduly pro jednotlivé důležité části aplikace. Následuje popis jednotlivých adresářů. Postupně od vygenerovaných po vlastní.

Kořenový adresář

V kořenovém adresáři se kromě zde vypsanych složek nachází také další pomocné soubory. `Gulpfile.js` používaný k nastavení chování `npm` při zapnutí lokálního vývojového serveru - nastavení automatické kompilace `scss` do `css` spolu s jeho konkatenací a minifikací, automatické obnovy stránek při změně v `template` souborech, automatické doplnění nového zkompilovaného `css` do stránky bez nutnosti jejího nového načtení a jiné.

Dále také `local.yml` a `production.yml` používané ke spuštění `docker-compose` a soubor `manage.py`, který slouží jako výchozí bod pro spouštění všech příkazů týkajících se `django`. Přes něj lze spustit lokální vývojový server, vytvářet databázové migrace, migrovat databáze, používat interaktivní `django shell` umožňující jednoduše pracovat s `django` modely v konzoli a další.

.envs

Tato složka slouží k ukládání lokálních a produkčních proměnných. Tyto soubory se běžně nenahrávají na žádný verzovací systém a nastavují se zvlášť na každém zařízení, jelikož obsahují citlivé přihlašovací a jiné údaje.

compose

Zde jsou uloženy soubory potřebné pro docker kontejnery. Podsložky `django`, `postgres` a `traefik` obsahují soubor `Dockerfile`, který při spuštění aplikace přes `docker-compose` nastaví příslušný kontejner.

config

Tato složka sdružuje hlavní aplikační nastavení. V souboru `base.py` se nachází všechna nastavení bez kterých by aplikace nemohla fungovat. Soubory `local.py` a `production.py` pak slouží k upřesnění některých nastavení pro produkční a lokální nasazení. Kromě toho se zde nachází také `urls.py`, umožňující tzv. `routing`, což je zjednodušeně řečeno určení co aplikace udělá při obdržení požadavku na určitou url adresu. Obecně se zde nastavují cesty k jednotlivým modulům aplikace do pole `urlpatterns`. Například jedna z položek

```
path("schuze/", include("volebni_kalkulacka.schuze.urls",
                        namespace="schuze")),
```

znamená, že pokud uživatel přistoupí na adresu `urlwebu.domena/schuze`, tak se všechny cesty budou hledat v souboru `urls.py` v modulu `volebni_kalkulacka.schuze`.

locale

Tento adresář je vytvořen aby obsahoval vygenerované jazykové soubory, pokud by aplikace používala více jazyků. `Cookiecutter-django` v základu podporuje multijazyčnost. V této práci, jelikož je specifická jenom pro Českou Republiku, tato možnost nebyla využita, nicméně pokud by byla potřeba, stačilo by pouze nastavit proměnnou `USE_I18N` v `base.py` na `True` a začít využívat tzv. `trans tagy` kolem každého textu zobrazovaného uživateli. Před nasazením aplikace do produkce by se spustil sběrač těchto textů, který by z nich vytvořil přehledný dokument. Zde by se texty ručně přeložily a následným kompilačním příkazem by se vytvořil binární soubor, ze kterého by `django` čerpalo při změně jazyka. Celý postup je přehledně popsán v dokumentaci².

requirements

Jelikož aplikace už v základu využívá několik python modulů, nalezneme zde jejich instalační seznam pro python instalátor `pip`. Je rozdělen stejně jako nastavení do 3 souborů - `base.txt`, `local.txt` a `production.txt`. Díky nim není nutné jednotlivé použité moduly složitě vyhledávat ručně, ale stačí spustit jeden příkaz a všechny moduly jsou nainstalovány.

volebni_kalkulacka

Hlavní aplikační složka. Zde budou umístěny všechny další navazující vytvořené moduly. V adresáři `static` se nachází všechny statické soubory, rozdělené podle typu - `CSS`, `fonty`, `obrázky`, `JavaScript` a `SASS`, před tím než se zkompiluje na `CSS`. Tyto soubory jsou po vykreslení webové stránky načítány z prohlížeče jako zdroje. Přímo z této složky je to však možné pouze v lokálním prostředí. Na produkci je nejprve nutné všechny statické soubory umístit na místo k tomu určené. `Django` to zvládne jedním jednoduchým příkazem³.

²Django překlady <https://docs.djangoproject.com/en/3.2/topics/i18n/translation/>

³Django statické soubory: <https://docs.djangoproject.com/en/3.2/ref/contrib/staticfiles/>

volebni_kalkulacka.templates

V `templates` se nachází HTML šablony používané návrhovým vzorem MVT. Jelikož šablony nejsou pouze samotné HTML, ale obsahují také Django Templating Language (django šablonovací jazyk), můžeme při vytváření šablon využívat určitý druh „dědičnosti“, kdy pro základní strukturu stránky, která se bude vždy opakovat vytvoříme `base.html` soubor. Zde se bude nacházet HTML hlavička, scripty, fonty a jiné statické věci využívané napříč celým projektem. Jednotlivé stránky pak dále budou tuto základní šablonu pouze rozšiřovat. To je umožněno prostřednictvím tzv. **block** (blokových) značek, kdy blok z rodičovské šablony je nahrazen blokem se stejným názvem ze šablony rozšiřující.

volebni_kalkulacka.users

Předpřipravený modul pro správu uživatelů. Vytváří rozhraní pro registraci, správu uživatelských účtů, změnu hesla, změnu emailu a potvrzení emailu.

dataImport

První z vlastních modulů. Slouží pouze k organizaci kódu potřebného pro stahování a aktualizaci dat z webu poslanecké sněmovny.

volebni_kalkulacka.psp_data

Tento modul slouží výhradně k organizaci souborů týkajících se dat z poslanecké sněmovny a djanga. Všechny modely pro výchozí data jsou definovány zde. Stejně tak je zde definován script pro prvotní stažení všech dat.

5.1.1 Import dat

Aby aplikace mohla fungovat, potřebuje mít přístup k aktuálním datům z poslanecké sněmovny. Ta jsou uložena s příponou `.unl` a jejich formát je podobný souborům `.csv` (**Comma Separated Values**), česky zvaným **čárkou oddělené hodnoty**. S tím rozdílem, že na rozdíl od CSV obsahují oddělovač i za posledního hodnotou na řádku, což standardu CSV neodpovídá [21]. Například tabulka `hl_poslanec`, uchovávající záznamy o hlasování jednotlivých poslanců pro každé hlasování má následující formát dat

```
1539|67018|A|
1537|67018|A|
1540|67018|A|
```

kde na prvním sloupci je odkaz na unikátní identifikátor poslance v tabulce `poslanec`, ve druhém sloupci je odkaz na unikátní identifikátor hlasování v tabulce `hl_hlasovani` a ve třetím sloupci je volba, kterou poslanec učinil.

Kromě toho jsem v průběhu vývoje narazil u datových souborů na několik dalších problémů.

- neexistující cizí klíče u několika tabulek
např. referencovaný nadřazený typ orgánu s ID 111 v tabulce `typ_organu`

- nekonzistence dat
V tabulce **zarazeni** může mít sloupeček **id_of** hodnotu cizích klíčů buď z tabulky **organy**, nebo z tabulky **funkce** v závislosti na hodnotě sloupce **cl_funkce**.
- data nekorespondující k deklarovanému typu podle popisu tabulek
Ve sloupcích podle popisu definovaných jako INT, čili celočíselné, se nachází nejenom čísla ale někdy také písmena či kombinace čísel a písmen. Například v tabulce **tisky** se ve sloupci **zm_lhuty** nachází kromě v popisu deklarovaných hodnot "1" a "2" také hodnoty "a" a "n".
- Kódování souborů nekorespondující s deklarovaným v popisu, ani s žádným z několika možných vyzkoušených.
Například u tabulky **tisky** by stejně jako u ostatních tabulek mělo fungovat deklarované kódování **windows-1250**, nicméně při pokusu o otevření s tímto kódováním dochází k chybám. Bylo proto nutné nastavit parametr **errors='replace'**, na základě kterého dojde k nahrazení nečitelných znaků. Tyto náhradní znaky však poté byly zobrazeny na webu a podstatně zhoršovaly uživatelskou zkušenost, proto bylo třeba je nahradit zpětně zase na odpovídající české znaky.
- Nejednoznačnost prázdných hodnot
V některých tabulkách je prázdné pole zaznačeno oddělovači sloupců u sebe, což odpovídá hodnotě NULL, čili žádné. Nicméně jinde jsou jako prázdné hodnoty použity mezery, či lomítka. Například v případě tabulky **osoby** a sloupce **pred** je to dokonce kombinace - buď NULL, nebo mezera.
- Nekonzistence názvů sloupců odkazujících na stejný prvek napříč tabulkami
V některých tabulkách je cizí klíč odkazující na volební období pojmenován jako **id_org**, v jiných zase jako **id_organ**, což velmi zneprůjemňuje práci s daty kvůli neustálé nutnosti kontrolovat názvy sloupců.

Aktualizace databáze

Aktualizaci databáze lze provést přes příkaz

```
python3 manage.py initial_import
```

z kořenového adresáře aplikace. Jeho definice se nachází v souboru `volebni_kalkulacka.psp_data.management.commands.initial_import.py`.

V prvotní verzi tento script⁴ zpracovával každý nahrávaný soubor řádek po řádku a každý zvlášť vkládal po dávkách do databáze. Tímto způsobem bylo možné lépe kontrolovat jak budou data zpracována a případně opravit nalezené chyby. Nicméně po změně koncepce aplikace a nutnosti zpracovávat všechna data bez omezení na pouze poslední volební období jsem narazil na výkonnostní problémy, kdy by zpracování zhruba 17 milionů řádků databáze přestalo být realizovatelné. Proto bylo nutné navrhnout lepší způsob.

Současná verze již využívá PostgreSQL příkaz **COPY**, díky kterému je možné zpracovávat celé soubory najednou [12], což velmi zásadně zvýšilo rychlost, jakou jsou data do databáze nahrána. Bohužel v důsledku této změny již není možné průběžně data upravovat. Navíc je třeba převést `unl` formát na `csv`.

⁴Script: krátký kus samostatně spustitelného kódu

Samotný script tedy funguje následovně. Nejprve jsou dynamicky získány odkazy všech stažitelných zip souborů na stránce s daty. Tímto zajistíme automatické přidání nových volebních období bez nutnosti úpravy kódu. Následně jsou staženy a rozbaleny.

Některé soubory zůstávají neměnné napříč všemi volebními obdobími. Jejich názvy jsou uloženy v souboru s konstantami v konstantě s názvem `INITIAL_FILE_NAMES` s datovým typem `dict` neboli česky slovník, kde je navázán název databázové tabulky na odpovídající jméno staženého souboru. Přes tyto dvojice (tabulka:soubor) je iterováno hned 2x. Poprvé je to kvůli již zmiňovanému `unl` formátu, konkrétně poslednímu oddělovači. Tento musí být odstraněn, jelikož příkaz `COPY` očekává, že za každým oddělovačem v souboru se nachází nový sloupec. Nahrávaný soubor je tedy otevřen a do nového souboru s příponou `_formatted` je uložen jeho obsah již bez tohoto nezbedného znaku.

Druhá iterace nejprve provede nad danou tabulkou příkaz `TRUNCATE RESTART IDENTITY`, který z ní smaže všechna data a vynuluje čítač řádků [13]. Dále získá hlavičky souboru, respektive tabulky, z konstanty `TABLE_HEADERS`, což je opět slovník dvojic, nyní ve tvaru (tabulka:pole názvů hlaviček). Následuje vypnutí databázových spouštěčů (anglicky `trigger`), aby bylo možné nahrát i data s nekonzistentními hodnotami, samotné nahrání dat a opětovné zapnutí spouštěčů.

Pro soubory s hlasováním je postup stejný, s tím rozdílem, že se budou objevovat nové názvy souborů pro nová volební období. Názvy jsou proto v této fázi vybírány dynamicky ze stažených souborů a všechny se nahrávají do jedné tabulky. Stejný princip je uplatněn jak pro tabulky `hl_poslanec` tak `hl_hlasovani`.

Po uložení všech potřebných dat jsou odstraněna zmatečná hlasování.

Dalším krokem je výpočet ohodnocení jednotlivých důležitých hlasování - těch která mají zásadní vliv na další existenci návrhů zákona. Tato funkce je jedna z nejpodstatnějších částí aplikace, jelikož je důležité uživateli hned zpočátku předložit v kalkulačce ty nejzajímavější možnosti. Samotný kód realizující tuto operaci se nachází v metodě `calculateHlasovaniRatings` třídy `DbManager` zajišťující práci s databází pro import dat, umístěné v souboru `volebni_kalkulacka/dataImport/Logic/DbManager.py`.

Nejdříve potřebujeme získat všechna důležitá hlasování za poslední dvě volební období. K tomuto účelu využívám SQL dotaz propojující tabulku `hl_hlasovani` s tabulkou `hist`, kde hledáme pouze hlasování spadající do ID orgánů značících poslední 2 volební období:

```
SELECT *
FROM
    psp_data_hl_hlasovani as hh

    INNER JOIN psp_data_hist as h
    ON h.id_hlas = hh.id_hlasovani

WHERE
    hh.id_organ IN (
        --last 2 election periods
        select
            id_organ from psp_data_organy
        where
            organ_id_organ is null
            and id_typ_organu = 11 --poslanecka snemovna
```

```

union
select id_organ-1 from psp_data_organy
where
    organ_id_organ is null
    and id_typ_organu = 11 --poslanecka snemovna
)

```

Výpis 5.1: SQL dotaz na všechna podstatná hlasování za poslední 2 volební období

Následně již v pythonu provedeme nad získanými daty cyklus, kdy pro každé z hlasování pokládáme další dotaz do databáze. Tentokrát hledáme dvojici (zkratka, výsledek), kde zkratka je zkratka strany a výsledek je hlasování poslance. Jedna dvojice za každého poslance zúčastněného v konkrétním hlasování.

```

SELECT
    zkratka,výsledek
FROM
    psp_data_hl_poslanec AS hp

    INNER JOIN psp_data_poslanec AS p
    ON hp.id_poslanec = p.id_poslanec

    INNER JOIN psp_data_zarazeni AS
z~ON p.id_osoba = z.id_osoba

    INNER JOIN psp_data_organy as
o~ON z.id_of = o.id_organ

    INNER JOIN psp_data_osoby as os
    ON os.id_osoba = p.id_osoba
WHERE
    hp.id_hlasovani = {hlasovani['id_hlasovani']}
    AND z.cl_funkce = 0 --clenstvi
    AND o.organ_id_organ = {hlasovani['id_organ']} --ID election period
    AND o.id_typ_organu = 1 --Klub
    AND ( --either membership is active (null)
        --or membership ended together with election period
        TO_DATE(z.do_o, 'YYYY-MM-DD') = TO_DATE(o.do_organ, 'DD.MM.YYYY')
        OR
        z.do_o IS null
    )
)

```

Výpis 5.2: SQL dotaz na získání dvojic zkratka, výsledek pro zadané hlasování

Výsledek databázového dotazu je načten do knihovny Pandas, což je rychlý, výkonný, flexibilní a jednoduše ovladatelný open-source nástroj pro manipulaci s daty a jejich analýzu [18]. Zde je opět cyklus, v tomto případě pro každou ze zúčastněných stran v hlasování. Výpočet ohodnocení hlasování se dá popsat sumou

$$OH = \sum_{i=1}^N \left(1 - \left| \frac{pocet_pro_i}{pocet_clenu_i} - \frac{pocet_proti_i}{pocet_clenu_i} \right| \right) * \frac{pocet_clenu_i}{pocet_vsech_zucastnenych} * 100 \quad (5.1)$$



Obrázek 5.1: Rozhraní pro uživatelské hlasování

kde OH je ohodnocení hlasování, N je celkový počet stran, i odpovídá jedné ze stran přes kterou cyklus zrovna prochází, $pocet_pro$ je počet poslanců ve straně hlasujících pro přijetí, $pocet_proti$ je počet poslanců ve straně hlasujících pro zamítnutí, $pocet_clenu$ je počet přítomných členů strany hlasujících pro přijetí/zamítnutí a $pocet_vsech_zucastnenych$ je celkový počet členů všech stran přítomných na tomto hlasování.

Cílem tohoto hodnocení je nalézt ta hlasování, kde poslanci jednotlivých stran hlasovali nejvíce rozdílně. Čili kde je nejmenší rozdíl mezi hlasy pro a proti uvnitř jednotlivých stran. Tento rozdíl dále násobíme poměrem velikosti strany k celkovému počtu členů parlamentu, aby se přizpůsobila váha jednotlivých stran k jejich velikosti. Výsledek hodnocení poté uložíme spolu s číslem hlasování do tabulky `hl_hlasovani_rating`. Při výpisu uživateli poté radíme hlasování od těch s největším ohodnocením.

Nakonec je spuštěna funkce na nahrazení nevalidních znaků popsanych v problémech s daty na začátku kapitoly 5.1.1.

Průběžná aktualizace dat bude prováděna pomocí `cronu`, což je standardní Unix nástroj používaný na naplánování automatického spuštění příkazů v zadaný čas [20]. Na serveru kde bude aplikace umístěna bude do `crontab` souboru třeba přidat záznam

```
0 3 * * * docker-compose -f /opt/volebni_kalkulacka/production.yml run
--rm django python manage.py initial_import
```

kde `/opt/volebni_kalkulacka/production.yml` bude odpovídat skutečnému umístění na serveru. V příkazu figuruje `docker-compose` příkaz, jelikož na produkci je aplikace umístěna v docker kontejnerech a je třeba tuto aktualizaci spustit uvnitř jednoho z nich.

5.1.2 Kalkulačka shod

U každého hlasování může uživatel sám hlasovat. K tomuto účelu slouží malý modul po straně, kde si může vybrat ze tří možností. Pro, proti a zdržet se. Může také svoji volbu zrušit. Vše je patrné z obrázku 5.1.

Své výsledky si uživatel může zobrazit v sekci Kalkulačka viz obrázek 5.2. Jelikož může hlasovat napříč několika volebními obdobími, byly výsledky rozděleny podle nich. Má tedy jeden výsledek pro každé období zvlášť.

Vyhodnocení shod probíhá v cyklu pro všechna období. Pro shodu s poslanci je použit SQL dotaz, který nejprve vyhledá všechny dvojice `matches`, `id_poslanec`, kde `id_poslanec`

Kalkulačka

Zde si můžete zobrazit své výsledky.
 Procenta udávají nakolik jsou Vaše volby shodné s jednotlivými poslanci a stranami v jednotlivých volebních obdobích. Čím více hlasování máte uskutečněno, tím přesnější vaše výsledky budou.

Aktualizovat výsledky kalkulačky

Poslanci		Strany	
Volební období 2017		Volební období 2017	
Ivan Adamec (ODS)	100.0%	SPD	52.6%
Hana Aulická Jirovcová (KSČM)	100.0%	Piráti	50.0%
Andrea Babišová (ANO)	100.0%	STAN	50.0%
Margita Balážiková (ANO)	100.0%	ODS	47.8%
Jan Bartošek (KDU-ČSL)	100.0%	KSČM	46.7%
Jan Bauer (ODS)	100.0%	TOP09	42.9%
Stanislav Berkovec (ANO)	100.0%	ČSSD	42.9%
Jaroslav Ržoch (ANO)	100.0%	ANO	42.3%
Lukáš Černohorský (Piráti)	100.0%	KDU-ČSL	40.0%
Jiří Dolejší (KSČM)	100.0%	Nezařaz	16.7%
Petr Dolínek (ČSSD)	100.0%		
Milan Feranec (ANO)	100.0%		
Eva Fialová (ANO)	100.0%		

Obrázek 5.2: Zobrazení výsledků kalkulačky v uživatelském profilu

je unikátní identifikátor poslance a sloupeček matches uchovává počet jeho shodných hlasování s uživatelem. Tyto informace jsou poté ve druhém databázovém dotazu spojeny do jedné tabulky spolu s daty o jednotlivých poslancích a jejich zařazení ve stranách v zadaném volebním období. Výsledkem je pole trojic `match_ratio`, `id_poslanec`, `zkratka`, kde `match_ratio` je poměr shod poslance s uživatelem a `zkratka` je zkratka jeho strany. Ten je poté zobrazen v kalkulačce. Výpočet shody uživatele s poslancem lze popsat vzorcem

$$SP_{ij} = \frac{pocet_hlasovani_i}{pocet_shodnych_hlasovani_{ij}} * 100$$
(5.2)

kde `SP` je procentuální ohodnocení shody poslance `j` a uživatele za volební období `i`, `pocet_hlasovani` je počet hlasování které uživatel uskutečnil spadajících do volebního období `i` a `pocet_shodnych_hlasovani` je počet hlasování, kde se uživatel a poslanec `j` ve volebním období `i` shodli.

```

DROP TABLE IF EXISTS t1;
CREATE TEMP TABLE t1 AS
    SELECT count(*) AS matches, id_poslanec
    FROM psp_data_hl_poslanec
    WHERE (id_hlasovani, REPLACE(REPLACE(vysledek, 'B', 'N'),'C', 'K'))
        IN %s
    GROUP BY id_poslanec
    ORDER BY matches DESC;

DROP TABLE IF EXISTS t2;
CREATE TEMP TABLE t2 AS
    SELECT DISTINCT ON (p.id_poslanec)
        (cast(t1.matches as decimal(7,2))/%s)*100
        AS match_ratio, t1.id_poslanec, o.zkratka,
        os.pred, os.jmeno, os.prijmeni, os.za
    FROM t1
        INNER JOIN psp_data_poslanec AS p
```

```

ON p.id_poslanec = t1.id_poslanec

INNER JOIN psp_data_osoby AS os
ON p.id_osoba = os.id_osoba

INNER JOIN psp_data_zarazeni AS z
ON p.id_osoba = z.id_osoba

INNER JOIN psp_data_organy as o
ON z.id_of = o.id_organ
WHERE z.cl_funkce = 0
AND o.organ_id_organ = {period_id}
AND o.id_typ_organu = 1
ORDER BY
p.id_poslanec, TO_DATE(z.od_o, 'YYYY-MM-DD') ASC
;

SELECT *
FROM t2
ORDER BY match_ratio DESC, prijmeni

```

Výpis 5.3: SQL dotaz na vyhledání shod uživatele s poslanci

Shody se stranami jsou počítány obdobně. Nejprve jsou získány počty poslanců ve všech stranách v procházeném volebním období. Dále počet hlasů shodných s uživatelem pro každou stranu. V posledním kroku se tyto dvě informace spojí a na výstupu je opět pole, tentokrát dvojic `match_ratio`, `zkratka`, kde `match_ratio` je procentuálně vyjádřena průměrná shoda uživatelských voleb s jedním poslancem ve straně. Výpočet lze popsat následujícím vzorcem

$$SS_{ij} = \frac{\frac{\text{pocet_shod_strany}_{ij}}{\text{pocet_odpovedi}_i}}{\text{pocet_clenu_strany}_{ij}} * 100 \quad (5.3)$$

kde SS je procentuálně vyjádřená uživatelská průměrná shoda s členem strany j ve volebním období i , `pocet_shod_strany` je počet shodných odpovědí s uživatelem za celou stranu j ve volebním období i , `pocet_odpovedi` je počet odpovědí uživatele spadajících do volebního období i a `pocet_clenu_strany` je počet členů strany j ve volebním období i .

```

--get last party for each member
DROP TABLE IF EXISTS osoby_zarazeni;
CREATE TEMP TABLE osoby_zarazeni AS
SELECT DISTINCT ON (os.id_osoba)
os.id_osoba, o.zkratka, z.od_o, z.do_o, o.od_organ
FROM
psp_data_osoby AS os
INNER JOIN psp_data_zarazeni AS z
ON os.id_osoba = z.id_osoba

INNER JOIN psp_data_organy AS o
ON z.id_of = o.id_organ

```



```

WHERE
    z.cl_funkce = 0
    and o.organ_id_organ = {period_id}
    and o.id_typ_organu = 1
ORDER BY
    os.id_osoba, TO_DATE(z.od_o, 'YYYY-MM-DD') ASC;

--get members count in party
DROP TABLE IF EXISTS tP1;
CREATE TEMP TABLE tP1 AS
SELECT
    COUNT(*) AS members_count, oz.zkratka
FROM
    osoby_zarazeni as oz
WHERE
    TO_DATE(oz.od_o, 'YYYY-MM-DD') = TO_DATE(oz.od_organ, 'DD.MM.YYYY')
GROUP BY
    oz.zkratka;

DROP TABLE IF EXISTS tP2;
CREATE TEMP TABLE tP2 AS
SELECT
    count(*) AS party_total_matches, oz.zkratka
FROM
    psp_data_hl_poslanec as hlp
    INNER JOIN psp_data_poslanec AS p
    ON p.id_poslanec = hlp.id_poslanec

    INNER JOIN osoby_zarazeni AS oz
    ON p.id_osoba = oz.id_osoba

--need to replace characters
--because they use multiple with same meaning
WHERE
    (id_hlasovani, REPLACE(REPLACE(vysledek, 'B', 'N'), 'C', 'K')) IN %s
GROUP
    BY oz.zkratka;

SELECT
    ((CAST(tP2.party_total_matches as decimal(7,2))/{answers_count})
    /tP1.members_count)*100 AS match_ratio, tP1.zkratka
FROM tP2
    INNER JOIN tP1
    ON tP1.zkratka = tP2.zkratka
ORDER
    BY match_ratio DESC, tP1.zkratka

```

Výpis 5.4: SQL dotaz na vyhledání shod uživatele se stranami



Obrázek 5.3: Email administrátorům s upozorněním na závadný obsah komentáře

5.1.3 Monitoring komentářů

Umožnění uživatelům přidávat obsah s sebou vždy nese rizika. V případě citlivých témat, jako je politika, jsou tato rizika násobně větší. Bylo proto třeba zavést určitá opatření pro umožnění alespoň částečné správy uživatelského obsahu.

Konkrétně byl přidán jednoduchý detektor vulgarit, kde se na základě seznamu sprostých slov převzatého ze stránky wiktioary.org [23] vyhodnotí jejich přítomnost v přidávaném komentáři a je zasláno upozornění do emailových schránek administrátorů. Ti poté můžou rychlým zákrokem komentář smazat a zabránit tak šíření závadného obsahu. Jak vypadá takové upozornění lze vidět na obrázku 5.3.

Tento způsob kontroly a správy byl zvolen z důvodu, že v případě například pouze blokování závadných slov by si kreativní uživatelé rychle uvědomili která slova algoritmus vyhodnotí jako závadná a našli si jiná, která by byla stejně vulgární, ale nedetekovaná. V případě povolení přidání všech komentářů a až dodatečné kontroly nemá běžný uživatel vůbec ponětí o detekci vulgarismů a nemá proto tendenci vymýšlet způsoby jak algoritmy obejít.

Tato kontrola může být v budoucnu dále rozšířena například na detekci odkazů na závadné internetové stránky, či blokování spamu.

Kapitola 6

Testování

Aby bylo možné aplikaci testovat, bylo nejprve nutné nasadit ji do produkčního prostředí a zpřístupnit uživatelům. K tomuto účelu jsem využil vlastní pronajatý VPS na Digitalocean.com. Aby uživatelé na nichž bude testování prováděno nemuseli psát do prohlížeče IP adresu, nasměroval jsem zde svoji doménu tomasmajerech.cz.

6.1 Průběžné testy

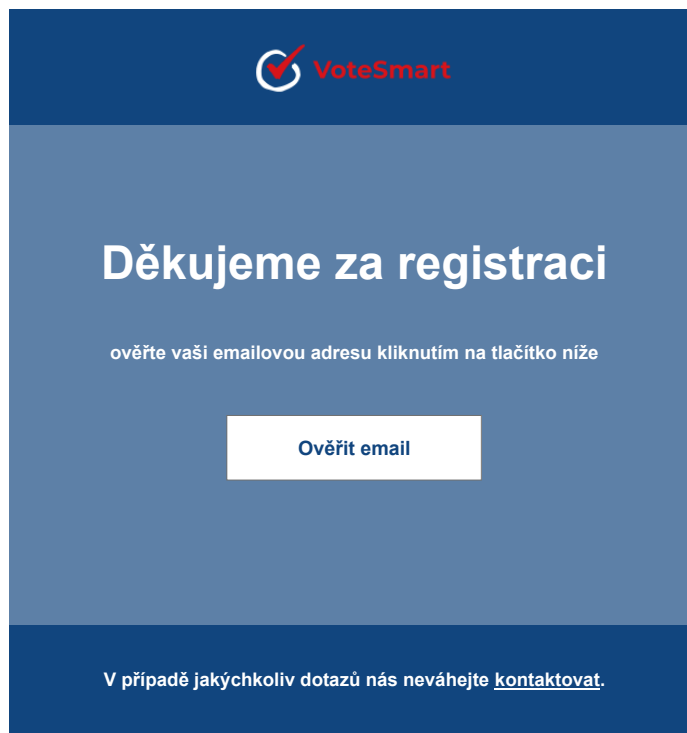
Testování aplikace probíhalo již od počátku. Drátový model byl po dokončení zaslán 6 nezávislým uživatelům ve věkovém rozmezí 16-57 let, kdy každý měl posoudit smyslnost rozložení dat a celkové struktury. Na základě jejich zpětné vazby byly provedeny drobné změny, například byl přidán obrázek České republiky na úvodní stranu a tlačítko kalkulačky posunuto ze středu více vlevo, či byl do hlasování přidán přehled hlasujících poslanců.

Druhé kolo testování proběhlo po dokončení grafického návrhu. Jeho cílem bylo ověřit, zda vizuální stránka aplikace nezaostává za běžnými standardy. Tento test musel být proveden 2x, jelikož v rané části vývoje došlo k celkové restrukturalizaci projektu a změně koncepce.

Z výsledků vyplynulo, že na všechny z testovaných subjektů celkové grafické provedení aplikace působí profesionálně, někdy až oficiálně v kontextu státní sféry. Na celkovém grafickém stylu tedy nebylo třeba nic upravovat. Nicméně objevily se výhrady k některým konkrétním částem aplikace, například k zobrazení jednotlivých hlasování k bodu pořadu schůze po najetí myši, viz obrázek 6.1, které bylo na základě tohoto hodnocení upraveno do současné podoby.



Obrázek 6.1: Původní návrh zobrazení hlasování pro bod schůze



Obrázek 6.2: Současná podoba emailových šablon

Po negativní reakci byly také upraveny šablony posílaných emailů pro registraci, změnu hesla a zapomenuté heslo. Z původní pouze textové podoby je nyní HTML emailová šablona, která působí na nové uživatele lepším dojmem. Aktuální podobu emailových zpráv, které uživatel obdrží, lze vidět na obrázku 6.2.

6.2 Testování produkční aplikace

Jelikož na testování produkční verze aplikace by již nebyl vzorek pouhých 6 uživatelů a jejich názorů dostatečný, bylo třeba prozkoumat nové možnosti. Tou první je tzv. **usability testing**, česky testování použitelnosti. Druhou je dotazník vytvořený pomocí Formulářů od Google.

Testování použitelnosti

Testování použitelnosti je způsob testování funkcionality webu, aplikace, či jiného digitálního produktu pozorováním skutečných uživatelů při jeho používání a provádění úkolů. Cílem tohoto testování je objevit oblasti, kde může dojít ke zmatení uživatele a odkrýt příležitosti k vylepšení celkového uživatelského zážitku [14].

Tato metoda je velmi časově náročná, jelikož vyžaduje plnou účast dvou a více lidí, kdy tester používá aplikaci, plní zadané úkoly a nahlas vyjadřuje svoje myšlenky, zatímco pozorovatel zaznamenává vše důležité.

Vzhledem k nutnosti osobní přítomnosti testovacích subjektů a současným platným omezením pohybu osob byl tento test aplikován pouze na osoby ve společné domácnosti. Konkrétně 3 muže ve věku 20-23 let bez předchozí zkušenosti s touto aplikací. Test byl pro-

váděn s každým subjektem zvlášť, nejprve na stolním počítači, poté na mobilním telefonu. V obou případech byly stejné úkoly.

1. Otevřít webovou stránku
2. Dát 5 hvězdiček a komentář Tomio Okamurovi
3. Najít hlasování které proběhlo naposledy
4. V tomto hlasování přidat svůj hlas
5. Najít 5 nejzajímavějších hlasování z pohledu kalkulačky a hlasovat i u nich
6. Prohlédnout si svoje výsledky v kalkulačce
7. Změnit heslo k účtu
8. Odhlásit se

Celkové výsledky testů dopadly spíše pozitivně. U žádného ze subjektů se neobjevily problémy s nalezením konkrétního poslance, komentováním, ani hvězdičkovým ohodnocením poslance. Při hledání posledního hlasování však shodně všichni uživatelé hledali ve špatné kategorii, kdy místo sekce Schůze, kde je poslední hlasování logicky umístěno v posledním bodě poslední schůze, hledali nejprve v sekci Hlasování, která však slouží k jinému účelu. Po otevření sekce Schůze již s nalezením nebyly žádné problémy, stejně tak s přidáním vlastního hlasu. Nalezení nejzajímavějších hlasování z pohledu kalkulačky již bylo pro testovací subjekty jednoduché, jelikož znali sekci Hlasování z předchozího kroku. Výsledky ke kalkulačce hledaly subjekty nejprve v modulu pro hlasování. Po neúspěchu však logickou dedukcí odvodili, že se nejspíše výsledky budou nacházet v profilu, kde je také našli. Změna hesla k účtu a odhlášení nepředstavovaly žádný problém.

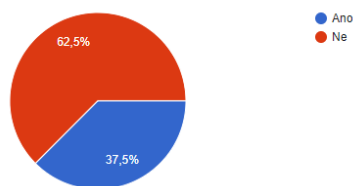
6.2.1 Testování s dotazníky

Kombinace dotazníku a odkazu na produkční aplikaci umožňuje rychle a efektivně získat názory velkého množství lidí. Tato metoda není zdaleka tak důkladná jako testování použitelnosti, nicméně umožňuje široký záběr a získání velkého množství informací a názorů.

Subjekty v dotazníkovém testování měly splnit stejné úkoly jako v testování použitelnosti a následně vyplnit formulář. Otázky ve formuláři byly:

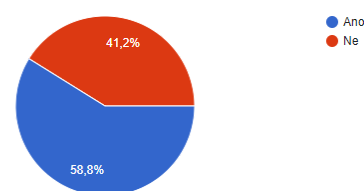
1. Použili jste k návštěvě webu mobilní telefon?
2. Podařilo se vám zvládnout všechny úkoly?
3. Pokud ne, u kterého úkolu jste se zasekl/a?
4. Pokud jste se zasekl/a, zkuste stručně popsat proč.
5. Jaké hodnocení byste aplikaci udělili?
6. Kdybyste mohl/a něco změnit, co by to bylo?

Použili jste k návštěvě webu telefon?
16 odpovědí



Obrázek 6.3: Statistika použití mobilního telefonu v dotazníku.

Podařilo se vám zvládnout všechny úkoly?
17 odpovědí



Obrázek 6.4: Statistika zvládnutí všech otázek v dotazníku.

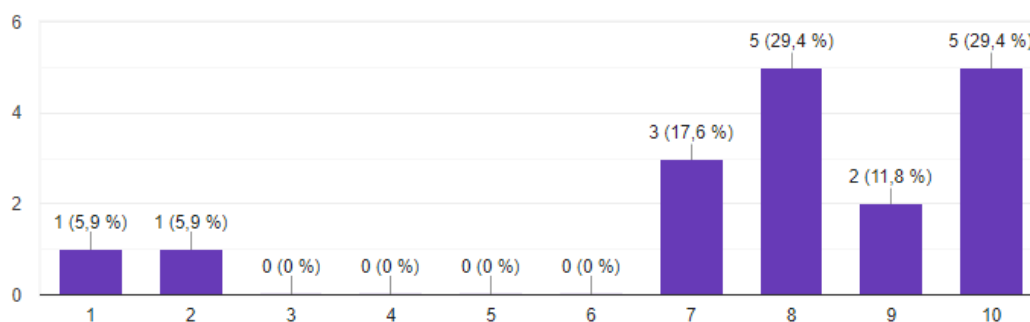
Dotazník byl osobně zaslán několika lidem a umístěn na FIT Discord server, nicméně nutnost postupovat podle úkolů v něm většinu lidí odradila od jeho vyplnění.

Celkem dotazník vyplnilo 17 lidí. Z nich 6 použilo k návštěvě webu mobilní telefon, viz graf na obrázku 6.3.

Většina, tvořená 10 uživateli, dokázala zvládnout všechny úkoly, viz 6.4. Z uživatelů s komplikacemi byli 4, kteří označili za blokující úkol č. 7, kde však kvůli chybně skrytému tlačítku nebylo tento úkol možné zpočátku splnit, proto tyto lidi zanedbám. Dále 5 lidí označilo jako bod záseku úkol číslo 2, nicméně dle navazujících textových odpovědí nebyla komplikace v nemožnosti bod splnit, ale v nechuti testovacích subjektů se do aplikace registrovat. Dle textových odpovědí by 3 uživatelé rádi zrychlili sekci Hlasování. Jedna z odpovědí je poměrně rozsáhlejší a upozorňuje na povinnost přidávat při registraci profilový obrázek a na nemožnost vyhledávání podle data či textu v sekci Hlasování, viz 6.6. Celkově uživatelé většinou hodnotí aplikaci nadprůměrně, viz 6.5

Jaké hodnocení byste aplikaci udělili?

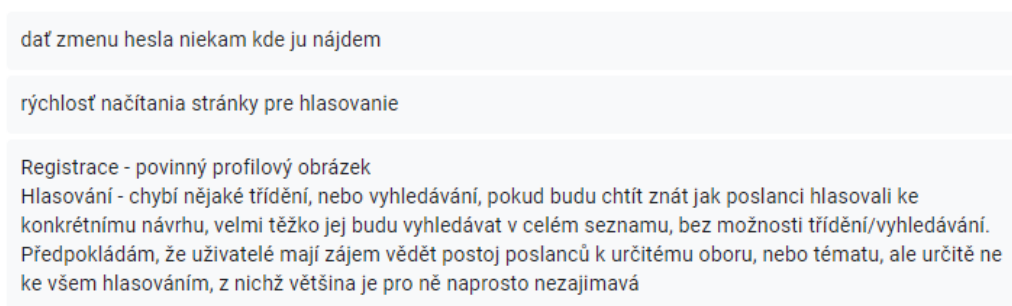
17 odpovědí



Obrázek 6.5: Statistika hodnocení aplikace



Obrázek 6.7: Filtr hlasování přidaný na základě zpětné vazby



Obrázek 6.6: Některé ze zajímavějších odpovědí v dotazníku k otázce co by uživatelé změnili

6.2.2 Závěry z testování

Na základě výsledků testů produkční aplikace byly provedeny následující úpravy:

- Výsledky kalkulačky byly přesunuty z uživatelského profilu do samostatné stránky s odkazem z hlavního menu. Díky tomu již nebude uživatel muset přemýšlet, kde má výsledky hledat.
- Tlačítko kalkulačky na úvodní straně nově odkazuje do sekce O nás, kde je použití kalkulačky vysvětleno, místo přechodu přímo na sekci Hlasování.
- V sekci O nás byla funkce kalkulačky důkladněji popsána.
- Načítání sekce Hlasování bylo zrychleno
- Byla odstraněna povinnost při registraci nahrávat profilový obrázek
- Bylo přidáno vyhledávání podle data a textu v názvu do sekce Hlasování. Vyhledání se uživateli zobrazí kliknutím na ikonu filtru. Na obrázku 6.7 je vyhledávací modul v otevřeném stavu.

Kapitola 7

Závěr

Cílem této práce bylo navrhnout a implementovat jednoduchý a efektivní nástroj, díky kterému si každý bude moci jednoduše porovnat svoje politické názory se skutečnými názory současných či bývalých členů Poslanecké sněmovny, který zároveň bude co nejvíce automatizován, aby úplně vymizela, či byla minimalizována nutnost lidského zásahu.

V teoretické části byly rozebrány základní pojmy nutné pro pochopení fungování Poslanecké sněmovny a legislativního procesu. Následně byla vyjmenována již existující konkurenční řešení a popsán formát dostupných dat na kterých bude práce stavět. Poté jsem se zaměřil na popis použitých technologií. Teoretickou část uzavírá rozsáhlá kapitola o návrhu aplikace, popisující jej od určení cílové skupiny a případů užití, přes vytváření modelů návrhů až po jejich skutečnou realizaci ve formě grafické předlohy. V závěru návrhové části jsou popsány některé teoretické aspekty důležité pro implementaci, například předpoklady k výběru důležitých hlasování, či zpracování výsledků v kalkulačce.

Na základě teoretické části poté vznikla funkční aplikace, jejíž nejdůležitější implementační detaily jsou popsány v praktické části práce a která by po nasazení na skutečný produkční server, a doplnění potřebného nastavení, měla být schopna samostatného fungování pro veřejnost. Tento předpoklad pramení z výsledků uživatelského testování, na základě kterého byly odstraněny všechny nalezené nedostatky.

Zůstal zde však velký nevyužitý potenciál pro budoucí rozšíření. Například by bylo možné vylepšit moderování uživatelských komentářů, vylepšit filtrování v sekci Hlasování, přidat nové možnosti pro registrované uživatele - například veřejný uživatelský profil a sdílení výsledků aplikace na sociální síť. Také by bylo možné přidat uživatelské hodnocení jednotlivých hlasování, podle kterých by mohlo jít následně hlasování řadit a mnoho dalšího.

Literatura

- [1] BROADBANDSEARCH.NET. *Mobile Vs. Desktop Internet Usage (Latest 2020 Data)* [online]. BroadbandSearch.net, 2020 [cit. 2020-11-22]. Dostupné z: <https://www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics#post-navigation-5>.
- [2] DATA, R. *HTML Introduction* [online]. Refsnes Data, 2020 [cit. 2020-11-22]. Dostupné z: https://www.w3schools.com/html/html_intro.asp.
- [3] DATA, R. *JQuery Introduction* [online]. Refsnes Data, 2020 [cit. 2020-11-22]. Dostupné z: https://www.w3schools.com/jquery/jquery_intro.asp.
- [4] DATA, R. *What is Bootstrap?* [online]. Refsnes Data, 2020 [cit. 2020-11-22]. Dostupné z: https://www.w3schools.com/whatis/whatis_bootstrap.asp.
- [5] DATA, R. *What is CSS?* [online]. Refsnes Data, 2020 [cit. 2020-11-22]. Dostupné z: https://www.w3schools.com/whatis/whatis_css.asp.
- [6] DATA, R. *What is npm?* [online]. Refsnes Data, 2021 [cit. 2021-5-1]. Dostupné z: https://www.w3schools.com/whatis/whatis_npm.asp.
- [7] FOUNDATION, P. S. *What is Python? Executive Summary* [online]. Python Software Foundation, 2021 [cit. 2020-11-22]. Dostupné z: <https://www.python.org/doc/essays/blurb/>.
- [8] GADAKH, N. *Why Django is so impressive for developing with PostgreSQL and Python* [online]. <https://www.fullstackpython.com/>, duben 2020 [cit. 2020-11-22]. Dostupné z: <https://www.enterprisedb.com/postgres-tutorials/why-django-so-impressive-developing-postgresql-and-python>.
- [9] GDJ. *[svg grafika]* [online]. Pixabay GmbH, říjen 2016 [cit. 2020-12-18]. Dostupné z: <https://pixabay.com/cs/vectors/%C4%8Desk%C3%A1-republika-zem%C4%9B-evropa-vlajka-1758820/>.
- [10] GREENFELD, D. R. *Deployment with Docker* [online]. readthedocs.io, 2021 [cit. 2021-4-28]. Dostupné z: <https://cookiecutter-django.readthedocs.io/en/latest/deployment-with-docker.html>.
- [11] GROUP, T. P. G. D. *PostgreSQL: The World's Most Advanced Open Source Relational Database* [online]. The PostgreSQL Global Development Group, 2020 [cit. 2020-11-21]. Dostupné z: <https://www.postgresql.org/>.
- [12] GROUP, T. P. G. D. *PostgreSQL: Documentation: 13: COPY* [online]. 2021 [cit. 2021-4-18]. Dostupné z: <https://www.postgresql.org/docs/13/sql-copy.html>.

- [13] GROUP, T. P. G. D. *PostgreSQL: Documentation: 13: TRUNCATE* [online]. 2021 [cit. 2021-4-18]. Dostupné z: <https://www.postgresql.org/docs/13/sql-truncate.html>.
- [14] HOTJAR. *A beginner's guide to user & usability testing* [online]. Duben 2021 [cit. 2021-4-21]. Dostupné z: <https://www.hotjar.com/usability-testing/>.
- [15] KOHOVOLIT.EU. *Info / VolebníKalkulačka.cz* [online]. KohoVolit.eu, 2006 [cit. 2020-11-22]. Dostupné z: <https://volebnikalkulacka.cz/info/>.
- [16] MAKAI, M. *PostgreSQL* [online]. <https://www.fullstackpython.com/>, 2021 [cit. 2020-11-22]. Dostupné z: <https://www.fullstackpython.com/postgresql.html#:~:text=PostgreSQL%20is%20the%20default%20database,to%20MySQL%2C%20SQLServer%20and%20Oracle.&text=PostgreSQL%20is%20an%20implementation%20of%20the%20relational%20database%20concept>.
- [17] NEHAMALI. *Top 10 Most Popular JavaScript Frameworks for Web Development* [online]. Geeksforgeeks, 2020 [cit. 2020-11-22]. Dostupné z: <https://www.geeksforgeeks.org/top-10-most-popular-javascript-frameworks-for-web-development/>.
- [18] NUMFOCUS. *Pandas* [online]. 2021 [cit. 2021-4-18]. Dostupné z: <https://pandas.pydata.org/>.
- [19] ODBOR INFORMATIKY, K. P. s. *Data Poslanecké sněmovny a Senátu* [online]. 2020 [cit. 2020-10-29]. Dostupné z: <https://www.psp.cz/sqw/hp.sqw?k=1300>.
- [20] PAIR NETWORKS, I. *What is Cron and How do I Use It?* [online]. Pair Networks, Inc., únor 2020 [cit. 2021-4-25]. Dostupné z: <https://www.pair.com/support/kb/configuring-cron/>.
- [21] SHAFRANOVICH, Y. *Common Format and MIME Type for Comma-Separated Values (CSV) Files* [Internet Requests for Comments]. RFC 4180. RFC Editor, October 2005. <http://www.rfc-editor.org/rfc/rfc4180.txt>. Dostupné z: <http://www.rfc-editor.org/rfc/rfc4180.txt>.
- [22] W3TECHS. *Usage statistics of JavaScript as client-side programming language on websites* [online]. Q-Success, 2020 [cit. 2020-12-4]. Dostupné z: <https://w3techs.com/technologies/details/cp-javascript>.
- [23] WIKTIONARY, K. autorů. *Kategorie:Vulgární výrazy/čeština* [online]. wiktionary.org, březen 2021 [cit. 2021-4-25]. Dostupné z: https://cs.wiktionary.org/wiki/Kategorie:Vulg%C3%A1rn%C3%AD_v%C3%BDrazy/%C4%8De%C5%A1tina.
- [24] ZADRAŽILOVÁ, T. *Senioři hledají na internetu praktické informace* [online]. Český statistický úřad, duben 2017 [cit. 2020-11-22]. Dostupné z: <https://www.statistikaamy.cz/2017/04/26/seniori-hledaji-na-internetu-prakticke-informace/>.
- [25] ČESKO. *Hlava druhá článek č. 15 ústavního zákona č. 1/1993 Sb.* 1993 [cit. 2020-11-16]. <https://www.psp.cz/docs/laws/constitution.html>.

- [26] ČESKO. *Hlava druhá článek č. 18 ústavního zákona č. 1/1993 Sb.* 1993 [cit. 2020-11-16].
<https://www.psp.cz/docs/laws/constitution.html>.
- [27] ČESKO. *Hlava druhá článek č. 39 ústavního zákona č. 1/1993 Sb.* 1993 [cit. 2020-11-16].
<https://www.psp.cz/docs/laws/constitution.html>.
- [28] ČESKO. *Hlava druhá článek č. 40 ústavního zákona č. 1/1993 Sb.* 1993 [cit. 2020-11-16].
<https://www.psp.cz/docs/laws/constitution.html>.
- [29] ČESKO. *§70 odst. 1 zákona č. 90 ze dne 19. dubna 1995 o jednacím řádu Poslanecké sněmovny.* 1995 [cit. 2020-11-16].
<https://www.zakonyprolidi.cz/cs/1995-90#p70>.

Příloha A

Instalační manuál

A.1 Lokální vývojové prostředí

Stručný návod jak spustit lokální vývojové prostředí. (psáno na OS Windows)

Další informace je případně možno nalézt také v Cookiecutter-django dokumentaci¹

1. Nainstalujte prerekvizity

Je třeba si nainstalovat všechny potřebné aplikace - python3, git, postgresQL databázi a Mailhog².

2. Vytvořte virtuální prostředí

Je dobrým zvykem umisťovat python projekty zvlášť, každý do vlastního virtuálního prostředí, aby se vzájemně nemohly ovlivňovat. Přejděte do složky kde budete chtít mít projekt umístěn a otevřete powershell. Nyní proveďte příkaz

```
$ python3 -m venv virtualenv
```

kterým je vytvořeno virtuální prostředí ve složce /virtualenv.

3. Aktivujte virtuální prostředí

Nyní je třeba toto prostředí v konzoli aktivovat pomocí příkazu

```
$ ./virtualenv/Scripts/activate
```

V tuto chvíli by měl být vidět název Vašeho virtuálního prostředí v konzoli. Odteď jsou implicitně všechny příkazy vykonávány uvnitř tohoto prostředí.

4. Naklonujte Github³ repozitář

Jelikož tato práce bude trvale umístěna na mém veřejném Github repozitáři, lze ho použít k naklonování souborů projektu. Spuštěním příkazu

```
$ git clone https://github.com/tmajerech/volebni_kalkulacka.git
```

se v aktuálním adresáři vytvoří podadresář volebni_kalkulacka, sloužící jako kořenový adresář aplikace a obsahující všechny potřebné soubory.

5. Nainstalujte potřebné python moduly

Následující příkazy změní složku do kořenového adresáře aplikace a spustí instalaci

¹Cookiecutter-django dokumentace <https://cookiecutter-django.readthedocs.io/en/latest/developing-locally.html>

²Mailhog: <https://github.com/mailhog/MailHog>

³Github: <https://github.com/>

python balíčků z instalačního seznamu. Díky tomu je není třeba instalovat po jednom ručně. Příkazem `upgrade pip` je také nejprve nainstalována poslední verze python instalátoru balíčků.

```
$ python3 -m pip install --upgrade pip
$ cd volebni_kalkulacka
$ pip install -r requirements/local.txt
```

Pokud aplikaci instalujete na operačním systému Linux, můžete v této fázi narazit na problémy s některými MYSQL balíčky. Následující kroky by mohly pomoci:

- `sudo apt-get install libmysqlclient-dev`
- Pokud po instalaci předchozí závislosti přetrvávají chybové hlášky jako například `No such file or directory #include "mysql/udf_registration_types.h"`, je možnost vyzkoušet smazání `"mysql/"` z hlavičkového souboru který tuto chybu způsobuje - cesta k němu by měla být viditelná v samotném chybovém hlášení. Tímto způsobem je možno odstranit i některé další chyby, ale toto řešení doporučuji zvážit pouze jako úplně poslední a dočasné.

6. Nastavte připojení k databázi

V kořenovém adresáři aplikace (`volebni_kalkulacka`) vytvořte soubor `.env` do kterého vložíte následující nastavení. Hodnoty upravte podle vlastních. Konstantu `USE_DOCKER` ponechte beze změny.

```
POSTGRES_HOST=localhost
POSTGRES_PORT=5432
POSTGRES_DB=volebnikalkulacka
POSTGRES_USER=postgres
POSTGRES_PASSWORD=root
USE_DOCKER=NO
```

7. Vytvořte databázové tabulky

Následující příkaz v kořenovém adresáři spustí tzv. **databázové migrace**, které vytvoří všechny potřebné tabulky podle vytvořených modelů.

```
$ python3 manage.py migrate
```

8. Spusťte lokální server

Příkazem

```
$ npm run dev
```

je spuštěn lokální vývojový server spolu s automatickým obnovením stránky při úpravě v HTML šablonách a také automatickou kompilací a doplněním CSS po uložení změn v SASS souboru - v tomto případě bez nutnosti obnovení stránky. Po startu serveru by se mělo automaticky otevřít okno s aplikací. V případě neotevření automaticky by se měla aplikace nacházet na url <http://127.0.0.1:8000/>.

9. Spusťte Mailhog lokální emailový server

Je třeba spustit stažený mailhog `.exe` soubor, který v příkazové řádce spustí lokální emailový server. Příkazový řádek je třeba nechat otevřený po dobu využívání emailů, jelikož nemá žádnou paměť a po vypnutí ztratí všechny zachycené emaily. Po zapnutí naleznete rozhraní emailového klienta na adrese <http://localhost:8025/>

10. Vytvořte administračního uživatele.
Například pro možnost mazat závadné komentáře je třeba vytvořit tzv. **super** uživatele. K tomu slouží příkaz

```
$ python3 manage.py createsuperuser
```


který už poté uživatele provede sám vytvářením.

A.2 Produkční prostředí

Další informace je možné opět nalézt v dokumentaci pro Cookiecutter-django⁴

1. Instalace prerekvizit
Je třeba nainstalovat Python3, Git, Docker⁵, Docker-compose⁶
2. Naklonování repozitáře
Ve vybraném adresáři proveďte příkaz

```
$ git clone https://github.com/tmajerech/volebni_kalkulacka.git
```


kterým naklonujete všechny soubory z githubu do nové složky `volebni_kalkulacka` v aktuálním adresáři.
3. Přidání enviromentálních proměnných pro django
Po přechodu do kořenového adresáře `volebni_kalkulacka` je třeba ve složce `.envs` vytvořit složku `.production` a v ní soubor `.django`. Do něj je vložte následující proměnné

```
# General
# -----

DJANGO_SETTINGS_MODULE=config.settings.production
DJANGO_SECRET_KEY=
DJANGO_ADMIN_URL=admin
DJANGO_ALLOWED_HOSTS=[]

# Security
# -----
# TIP: better off using DNS, however, redirect is OK too
DJANGO_SECURE_SSL_REDIRECT=False

# Email
# -----
DJANGO_SERVER_EMAIL=
DJANGO_EMAIL_HOST_USER=
DJANGO_EMAIL_HOST_PASSWORD=
```

⁴Cookiecutter-django produkční nastavení: <https://cookiecutter-django.readthedocs.io/en/latest/deployment-with-docker.html>

⁵Docker instalace: <https://docs.docker.com/get-docker/>

⁶Docker-compose instalace: <https://docs.docker.com/compose/install/>

```
# django-allauth
# -----
DJANGO_ACCOUNT_ALLOW_REGISTRATION=True

# Gunicorn
# -----
WEB_CONCURRENCY=4

# Redis
# -----
REDIS_URL=redis://redis:6379/0
```

a doplňte všechny chybějící proměnné. Do `DJANGO_ALLOWED_HOSTS` přijde název produkční domény, případně také její IP adresa a další povolené a url. Nastavení emailu je možné simulovat například přes osobní gmail účet s povoleným připojením třetích stran.

4. Přidání enviromentálních proměnných pro postgresSQL

Opět ve složce `.production` je třeba vytvořit soubor `.postgres` a do něj vložit

```
# PostgreSQL
# -----
POSTGRES_HOST=postgres
POSTGRES_PORT=5432
POSTGRES_DB=volebni_kalkulacka
POSTGRES_USER=vjZtIkTyKgMaZGLVWHVlySsL1lRCbGVx
POSTGRES_PASSWORD=
    LGpkCqeeUrvzvonMonlgeSa11EBqxOM8IAZpyzNuDb7exKxrIVZZLqkYDGTtH15s
```

kde text za proměnnou `POSTGRES_PASSWORD` bude na jednom řádku, zde je na samostatném kvůli šířce stránky. Tyto údaje byly vygenerovány při vytváření šablony aplikace v Cookiecutter-django a mělo by je jít kdykoliv nahradit novými údaji korepondujícími s nastavením postgresSQL kontejneru.

5. Nastavení SSL

SSL, neboli **Secure Sockets Layer** (vrstva bezpečných socketů) je bezpečnostní standard pro vytváření šifrovaného spojení mezi serverem a klientem, v našem případě konkrétně mezi webovým serverem (stránkou) a prohlížečem. Nemít HTTPS znamená, že je možné z komunikace odposlechnout například přihlašovací údaje [10].

V této aplikaci SSL zajišťuje služba Traefik. Mně se bohužel nepodařilo tuto službu spustit, protože jsem ji odkládal až úplně nakonec, ale nemělo by to být složité. Postupujte podle Cookiecutter-django manuálu⁷

6. Sestavení docker kontejnerů

V kořenovém adresáři aplikace spusťte příkaz

⁷Cookiecutter-django SSL manuál: <https://cookiecutter-django.readthedocs.io/en/latest/deployment-with-docker.html>

```
$ docker-compose -f production.yml build
```

kterým se sestaví všechny potřebné docker kontejnery. Příkazem

```
$ docker ps -a
```

lze zobrazit všechny vytvořené kontejnery včetně neaktivních.

7. Spuštění aplikace

Opět v kořenovém adresáři spusťte příkaz

```
$ docker-compose -f production.yml up
```

kterým by se měly spustit všechny potřebné docker kontejnery a aplikace by měla sama začít poslouchat na portu 80 pro http a 443 pro https.

8. Nainportujte data

Spusťte příkazem

```
$ docker-compose -f production.yml  
run -rm django python manage.py initial_import
```

import dat do databáze. Poté nastavte cron, který bude tento příkaz spouštět pravidelně v noci. Tato funkce je náročná na výkon serveru, takže doporučuji spouštět když nebude server zatížený. Vzor nastavení cronu můžete převzít z kapitoly [5.1.1](#).

Pokud tuto práci někdo bude dále rozšiřovat a narazí v průběhu na nečekané komplikace, může mě kontaktovat a jsem ochoten, pokud to bude v mých silách, pomoci s nastavením a pochopením.